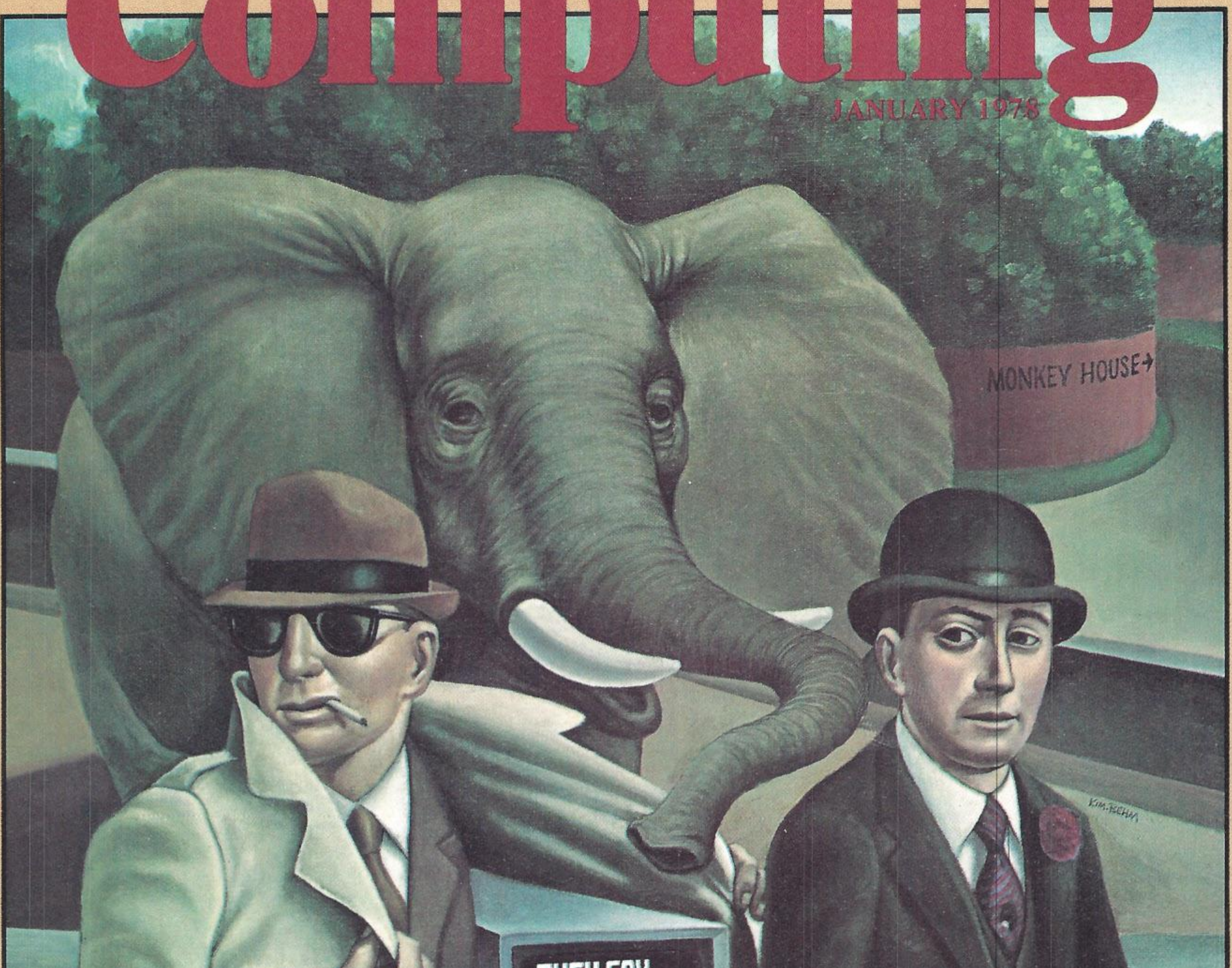
$2.00

Personal Computing

JANUARY 1978



Cloak and Dagger
with your personal computer

Computer Mastery • Magic for Educators • Computer Boxing

Verbatim™

The whole message is quality.

You want to record your message verbatim—word for word—whether it's bits, bytes or "Dear Folks" translated into word processor language.

Our objective in manufacturing recording media for the electronics industry—digital tape cassettes, floppy disks, mag cards, computer cartridges—is to give you the finest, the best, the most dependable, the most cost-effective.

That means rugged, long-lived, abrasion-resistant recording media with superior magnetic qualities. If we made tires, they'd be steel-belted radials.

We delivered our first digital grade certified tape cassettes back in the beginning, 1969. We made the first commercial 3740-compatible floppy disks that didn't bear IBM's name. And the first Flippy® reversible flexible disks with *anyone's* name on them. The first mini data cassette is ours. And we've got the newest miniature flexible disk, the MD 525.

Now, **Verbatim** media. It's a new formulation of ferric oxides, an advanced macromolecular binder system to adhere it to the tough polyester film, and a process control system that demands over 200 separate quality checks before the material is cut, packaged, and certified to be 100% error-free.

The final quality check? "Make it pretty!" Our production people tell us that magnetic recording media is one of the rare instances in manufacturing where aesthetic appearance translates directly into final product quality. It has to look beautiful to work beautifully.

We have the formulas, the machines, the technology to make high quality recording media. But it takes the best people in the industry to deliver Verbatim disks, cards, cartridges and cassettes. You'll find them at your favorite retail computer store.

Information Terminals Corp.®
Sunnyvale, California

Welcome!

Welcome to the new PERSONAL COMPUTING. We have an enlarged staff, an enlarged format, an enlarged outlook and an enlarged office. We've moved our Albuquerque office to Boston to bring our publishing, editorial, art and production offices under one roof and enhance the togetherness of the PC staff with the staff of our other publications. And starting with this issue, you'll be receiving PERSONAL COMPUTING every month instead of every other month. Our increased publication schedule is going to make it possible for more of you to tell us what you're doing, how you are doing it, and why you're doing it.

Yes, Virginia, you too can write for PERSONAL COMPUTING. You'll find out how on page 111 of this issue. Here's a sampling of some of our current authors.

Ed Bernstein (author of *Banner Fun*): "I began working with computers as a freshman, and expect to cease work about two years after my death. I'm president of my high school's computer club."

Ramond Howell (*Scaling the Cliffs*): "I'm a missionary serving the Jungle Aviation and Radio Service (JAARS), the transportation and communication arm of Wycliff Bible Translators whose task is to get God's Word, the Bible, into previously unwritten languages of every person on earth."

Ken Fritz (*Balanced Portfolio*):"I'm an engineer by trade, but a personal computer enthusiast by inclination. My interest in games programming dates back to the battle days of megabuck mainframes. I'm configuring a system for my 6- and 4-year-old daughters, but my third and youngest daughter will have to wait for a hand-me-down from her sisters."

Barbara Tutt (*Miracle of Babbage Eve*): "I'm a linguist turned freelance writer, and a computer hobbiest by marriage. I hope to use the computer I'm assembling with my husband for speech synthesis and word processing."

Joseph Roehrig (*Computer Boxing*): "I'm the manager of financial evaluation and budgets at the National Broadcasting Company in New York. A small ad in the New York Post got me involved in microcomputers, so I bought a computer kit when my wife was expecting our first child. Now Tricia-Anne is nine-months-old and enjoys banging on the computer keyboard.

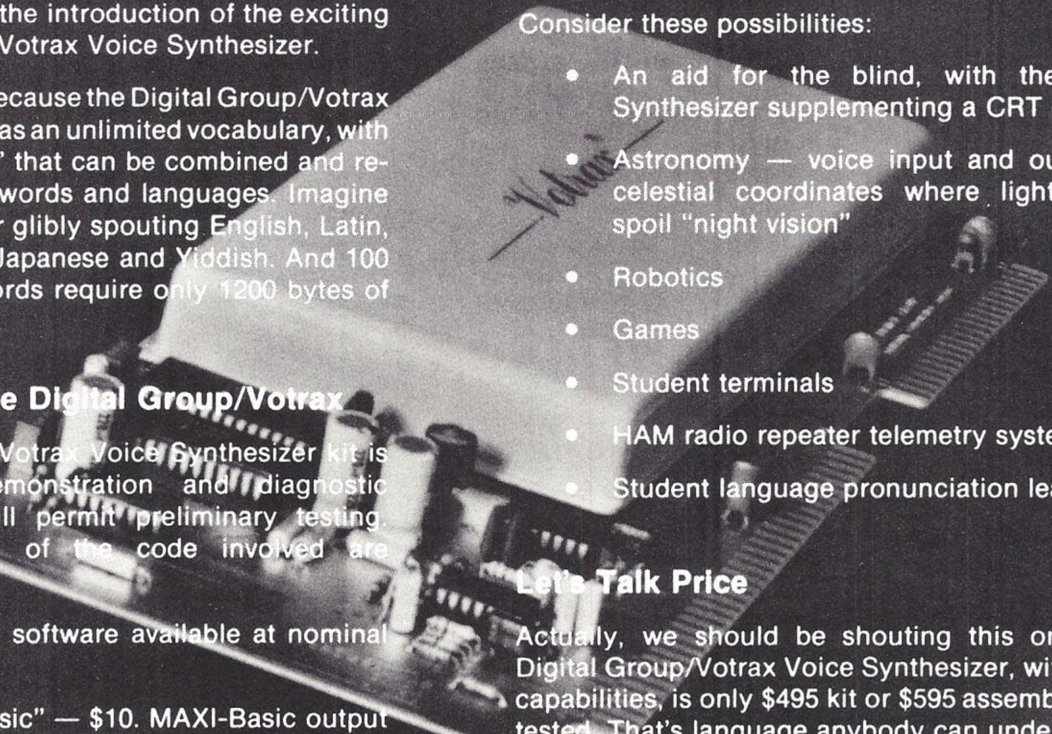
See, we weren't kidding. Our authors come from a variety of backgrounds. So when you finish reading our welcome, if you're interested in writing, take a look at the guidelines for prospective authors on page 111 and join the crowd.

Next month in PERSONAL COMPUTING we'll tell you how one young man built a robot and programmed it using BASIC, and how to snatch a tidy income printing amortization tables. We'll take you on a stroll through the intricacies of the microcomputer industry, let you in on the best deals on RAM memory, uncover the latest thrills (and pitfalls) of buying Radio Shack's TRS-80 and Commodore's PET, and teach you a cheap way to computerize all your household appliances (even if you don't know the difference between an amp and an ohm). Plus lots, lots more that you'll have to wait until next month for – because we promised not to tell.

The Editors

Talk to our Computer... and it will talk back!

(Plainly speaking, it's only from the Digital Group.)



Now, your Digital Group computer becomes more than a silent partner. You can vocally command your computer . . . it will listen . . . and it will talk back to you. How? With the introduction of the exciting new Digital Group/Votrax Voice Synthesizer.

All this is possible because the Digital Group/Votrax Voice Synthesizer has an unlimited vocabulary, with 64 "human sounds" that can be combined and re-combined to form words and languages. Imagine your own computer glibly spouting English, Latin, Spanish, Russian, Japanese and Yiddish. And 100 average English words require only 1200 bytes of memory!

Programming the Digital Group/Votrax

The Digital Group/Votrax Voice Synthesizer kit is supplied with demonstration and diagnostic software which will permit preliminary testing. Assembler listings of the code involved are included.

We have additional software available at nominal cost:

- "Talking Basic" — $10. MAXI-Basic output converted to English.
- "Talking CW" — $10. For impressing your HAM buddies. Requires the forthcoming HAM interface card.
- "Latin and Spanish Talking" — $10. Hear the computer repeat letters and words typed in Latin or Spanish.
- Demonstration Tape — $5. A sample of audio tape and a complete explanation of the system.

Bonus: A basic input circuit is included that may be programmed to understand a small vocabulary of voice commands.

Unlimited Applications

Consider these possibilities:

- An aid for the blind, with the Voice Synthesizer supplementing a CRT display
- Astronomy — voice input and output of celestial coordinates where light would spoil "night vision"
- Robotics
- Games
- Student terminals
- HAM radio repeater telemetry systems
- Student language pronunciation learning

Let's Talk Price

Actually, we should be shouting this one. The Digital Group/Votrax Voice Synthesizer, with all its capabilities, is only $495 kit or $595 assembled and tested. That's language anybody can understand.

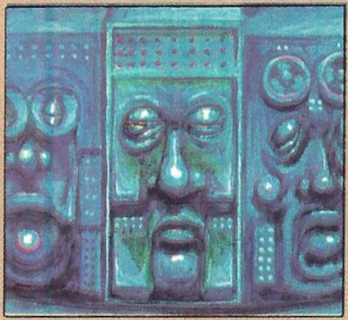
O.K., you've listened briefly to what we have to say about the new Digital Group/Votrax Voice Synthesizer. But we can keep right on talking! Write or call today for *all* the details — music to your ears.

the digital group inc.

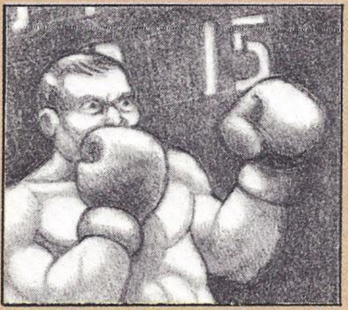
po box 6528 denver, colorado 80206 (303) 777-7133

January 1978

VOLUME II, NO. 1



BABBAGE EVE 55



BOXING BY COMPUTER 88

Cover Illustration by Kim Behm

LAUNCHING PAD

ON THE LIGHTER SIDE

IN THE MONEY

ONCE UPON A TIME

Subscription rates. U.S.: 1 year (12 issues) $14; 2 years (24 issues) $26; 3 years (36 issues) $38. Canada & Mexico: add $4/year for surface mail, $8/year for airmail. In Japan/Asia: Personal Computing, CQ Magazine, 11-8, Narita-Higashi 1-Chome, Suginami-KU, Tokyo 166, Japan. All other countries: add $8/year for surface mail, $36/year for airmail. Send subscription orders and changes of address to: Circulation, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215. **Back issues.** U.S.: $2. Canada & Mexico: $3. All other countries: $5.

Publisher. Published monthly by Benwill Publishing Corp.: Harold G. Buchbinder, Chairman of the Board; George Palken, President; Esther Shershow, Treasurer. Executive, Editorial and Subscription Offices: 1050 Commonwealth Ave., Boston, MA 02215. Controlled Circulation postage paid at Long Prairie, MN. Membership in Audit Bureau of Circulation pending.

DEPARTMENTS

Enough, enough!

Mr. Overington's article, "Chess from an Information Storage Angle," in the September/October issue, prompted me into another brief round of "the big numbers game."

We can place K distinct objects on 64 squares in (64!)/(64 - K)! ways. For our purposes, K can be any number between 32 and 2 (you'll always have 2 kings on the board) giving a grand total of (64!/32!) + (64!/31!) + . . . + (64!/61!) + (64!/62!). Using the first four terms I get 64!((1/32!) + (1/33!) + (1/34!) + (1/35!)) = (1.27 * 10^{89}) (3.92 * 10^{-36}) = 4.97 * 10^{53}; doubling gives 9.95 * 10^{53} – just short of 10^{54}.

For the lower bound, we can start by placing 32 identical pieces on the board in 64!/((32!) (32!)) = 1.83 * 10^{18} ways. Considering two classes of pieces (black & white) gives us 64!/((16!) (16!)(32!)) = 1.1 * 10^{27}. Each of those classes has two subclasses (pieces & pawns) which gives us 64!/($(8!)^4$ (32!)) = 1.8 * 10^{35}. The problem with pawns is that they are not permitted on the first rank. To make things easy for myself, I'll also disallow them on the eighth rank; then we have 45! ((8!) (8!) (32!)) = 2.9 * 10^{16} ways of putting the pawns in the middle six ranks. There is 1 king, 1 queen, 2 bishops, 2 knights, and 2 rooks, giving us 48!/($(1!\ 1!\ 2!\ 2!\ 2!)^2$ (32!)) = 7.37 * 10^{23} ways of putting the pieces on the remainder of the board, for a grand total of 2.14 * 10^{40}. Doubling to account for either player having the move gives 4.28 * 10^{40}. And that's for 32 men only. For 31 men . . . . . Enough, enough! I'm not yet that ambitious.

P.S. The new high needs 180 bits just to count that high; the lower bound needs 135.

Leigh Janes
Rocky Hill, Connecticut

Happy using logarithms?

I have never been satisfied with the very few digits of accuracy in logarithms. For some applications they are sufficient. For many others, not.

Recently I became disgusted using logs to calculate roots of numbers. I did not know any other practical method, so I sat down and invented one.

I call it a generalization of Newton's Method for calculating square roots:

$$X_{n+1} = (X_n + A/X_n)/2.$$

(Newton's Method For Finding $(A)^{1/2}$.

For fast convergence to many digits of accuracy, you should use as first approximation an integer which is very close to the square root.

To shorten my story, I fiddled around and came up with:

$$X_{n+1} = ((P-1)X_n + T/X_n^{P-1})/P.$$

(Corson's Method For Finding $(T)^{1/P}$.

It *seems* to work; at least it does for the examples I have tried. Starting with X_O, in those examples, X_4 yielded 7 digits of accuracy: the 5th root of 2 is 1.1486983; and the 3rd root of 9 is 2.0800838. I would suggest you would be wasting computer time in applying this method to calculating fret positions on 31-tone guitars, etc., since the required accuracy there is but 0.1 mm (rounded from hundredths).

I can hear you saying: "With that many digits accuracy, he sure didn't use BASIC!"

I used a $10 TEAL (Japanese make) pocket calculator. Next, I shall write the program into BASIC, and sneak into the nearby friendly (only if you buy something!) BYTE Shop.

Since I invented (or possibly rediscovered) this algorithm, I wish it to be released freely to my fellow computer hobbyists – and its use not hamstrung by some commercial outfit.

If anyone can improve upon this little gem, and cause faster convergence, I would welcome it! Even if you use orthogonal functions!

Clyde Eugene Corson, M.S.
Lawndale, California

What you call "Newton's Method" and "Corson's Method" are both examples of a more powerful formula:

$$X_{n+1} = X_n - F(X_n)/F'(X_n)$$

That powerful formula solves the equation F(X)=0, where F can be "practically any" function. In the formula, F′ means the "derivative" of F, in the sense of calculus. To prove Corson's Method from the powerful formula, define F(X) to be X^P-T. To find out more about the powerful "Newton-Raphson" formula, read any book on "numerical analysis" or "numerical methods."

Wanted: hardware evaluation

Keep your eyes
 on the threes and fives:
As everyone knows
 they have petals around the rose.

I'm enjoying your magazine cover to cover. I would like to see some articles evaluating hardware and systems in layman's terms. This would require standards (speed, capacity, flexibility, special features, compatibility) with which to compare, and might raise some objections from advertisers, but a fair, objective report will sell more computers and provide a valuable service to those of us who are 'shopping.'

Kerry McCalla
Birmingham, Alabama

If any of you readers wants to team up with a friend and write an article comparing your computers, we'll buy it, check its accuracy, and run it.

No petals around the rolls?

About the petals around the rose in the Sept./Oct. issue – seems to me that having an answer of 20 has a probability of one in 7776 rolls.

John Ramsey
Rochester, New York

Oh!

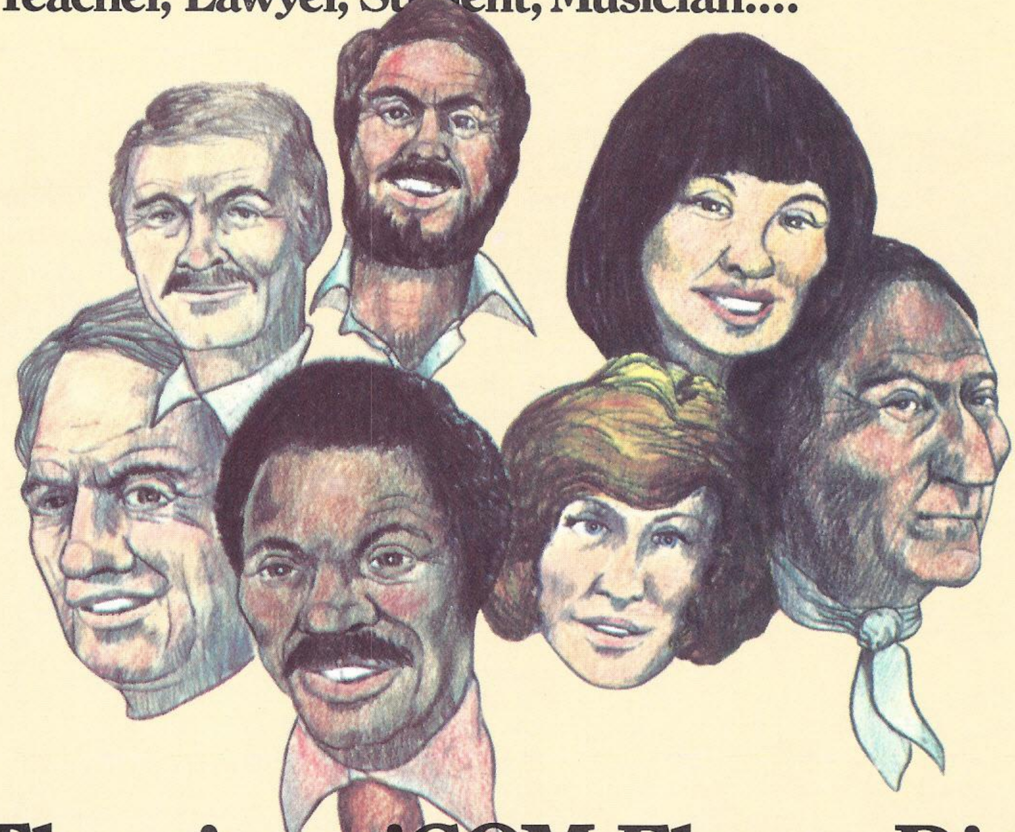
Programmer licensing: a definite 'no'

When I was a child my friends and I had a nifty clubhouse in the backyard. On the door we put a great big "KEEP OWT!" sign. In the years since, my ego has become stronger. As an adult, I no longer fear other people and do not feel the need to put them down by excluding them.

If my experience is not unique then there is hope for David Weinglass, whose letter is in the September/October issue, and the person I overheard in Atlanta who said: "We don't want computers to be so simple that just *anybody* can use them."

Although their reasoning may be obscure, the motive is not: "Knowledge

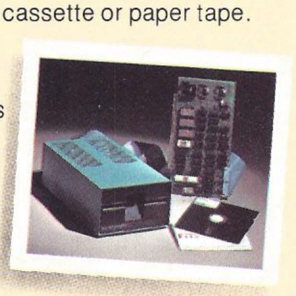
Rich Man, Poor Man, Merchant, Physician, Teacher, Lawyer, Student, Musician....



There's an iCOM Floppy Disk System for Everyone!

More Uses
People from every walk of life are adding iCOM® Floppy Disks to their microcomputers for such diverse tasks as payroll, inventory control, mailing lists, game playing, record keeping, parts ordering . . .
We've uncovered some innovative applications, too: The sailboat architect who puts equations and algorithms on an iCOM disk to test his nautical theories; the student who has automated a bowling alley; the iCOM dealer who designed an environmental control system for a university.

More Speed
These users have found iCOM floppies to be much faster and more versatile than cassette or paper tape. With iCOM, programs can be loaded in seconds; files updated in minutes; hundreds of programs can be stored on a single disk.

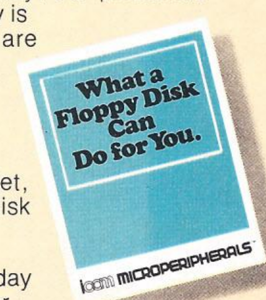


More Models
iCOM has Frugal Floppies™, Dual Floppies, Microfloppies™ (using the new 5 ¼" diskette), and other new approaches to floppy disk systems. Each is hardware and software compatible with Altair™, IMSAI, Poly 88, Sol-20 and other microcomputers using the Altair S-100 bus format.

More Software
Then there's iCOM's famous software: Powerful field-proven FDOS-II with macro-assembler, string-oriented text editor, and file manager. Plus easy-to-use compatible 8K Disk BASIC. Each with super features such as: named variable length files, auto-file create, open and close, multiple merge and delete . . . and more.

More Backup
We've been building floppies for microcomputers for more than 3 years. Long before the rest. Thousands of systems are operating perfectly in the field. And we're part of Pertec Computer Corporation, one of the largest manufacturers of peripherals, microsystems, data entry products and data processing systems. We'll be around whenever you need us.

More Dealers
Maybe not in quantity, but in quality. We've chosen our dealer network carefully to assure you of assistance every step of the way. Our prices are right. Our delivery is fast. Our dealers are experienced and knowledgeable.



Must Reading
Our free booklet, "What a Floppy Disk Can Do for You" is must reading. Send for yours today or visit your dealer.

icom® MICROPERIPHERALS®

a division of Pertec Computer Corporation

6741 Variel Ave., Canoga Park, CA 91303 U.S.A.
Tel. (213) 348-1391 TWX 910-494-2788

is power and so it tends to be hoarded. Experts in any field rarely want people to understand what they do, and generally enjoy putting people down." –T.H. Nelson in *Computer Lib.*

The ideas in Weinglass's letter, like evaluating "the quality of the people" – not the *programs,* mind you, the *people* – are downright spooky.

As a would-be scientist, I can see enough problems in this world to require the collective genius of *all* of us. I see nothing to be gained by disposing of a large proportion of a valuable natural resource – people. That is why I believe I can "really contribute to society" by what I am doing – teaching everyone who will listen all I can about computers.

If your magazine upsets the "KEEP OWT!" crowd, then it is my kind of magazine. Please begin my subscription now.

J.P. Peer
Berne, Indiana

• • •

I have just completed reading Mr. Weinglass' letter (September/October 1977) and have to reply.

I, too, am a programmer and proud of my work. But if all programmers thought as Mr. Weinglass does, computing would be one of those dark arts available to the select few and understood by only half of those.

I can detect no lack of respect in your magazine; I just feel sorry that Mr. Weinglass is so obsessed with the existing hierarchy that he can't see the new micro movement growing. He should not be afraid to use computers in new and imaginative ways. How else can we increase our knowledge of these wonderful machines? Also, the licensing of programmers, to which Mr. Weinglass refers, will raise computer fraud to higher heights – only the really good *licensed* crooked programmer will get hired.

I don't think that the "lemonade operator" will actually try to market his services as much as he will try to learn some new techniques to amuse himself. After all, with the crooked licensees running around, who will have the money to hire an independent operator?

Richard W. Crooks
Belvidere, Illinois

About David Weinglass' letter to the editor in the September/October issue – I too am a professional programmer with many years' experience but with no stomach for the snobbish pseudo-intellectual approach that Mr. Weinglass has taken.

"Anybody" does not have to be "allowed to use computers any way he wants to." Computers are for the "elite," right? *Wrong!!!*

Good programming habits are a matter of education and Mr. Weinglass' legal standards will do the same thing that all such licensing does: spawn mediocrity.

Good programmers, however, who are willing to educate and set an example of a high goal to shoot for will have a far more meaningful and helpful effect on the general level of quality in programming than all the legal standards that Mr. Weinglass and his ilk can dream up.

The idea that one would have to prove to authorities that he can contribute to society before being allowed to practice his profession was prevalent in Nazi Germany.

Leland C. Sheppard
Sunnyvale, California

• • •

I couldn't help responding to the ominous tone of reader Weinglass's letter concerning licensing computer programmers (Sept./Oct. 1977). Licensing seems to be a perennial method used by long established people practicing a profession to prevent the entrance into the field by new competition. They always try to justify it by saying that licensing will prevent the fly-by-night type of operator. One wonders, though, if it is so difficult for firms purchasing computer services to develop standards to judge them, then why should we be so trusting of an "authoritative" group that can *impose* standards on everyone (backed by the force of government). The fact is we can choose between a system where some individuals and firms will no doubt be cheated by bad programmers and one where there is a market open only to those approved by a monopolistic standards agency. I fear the benefits of the latter system will be far outweighed by the harm.

S.L. Shakocius
Palos Verdes Estates, California

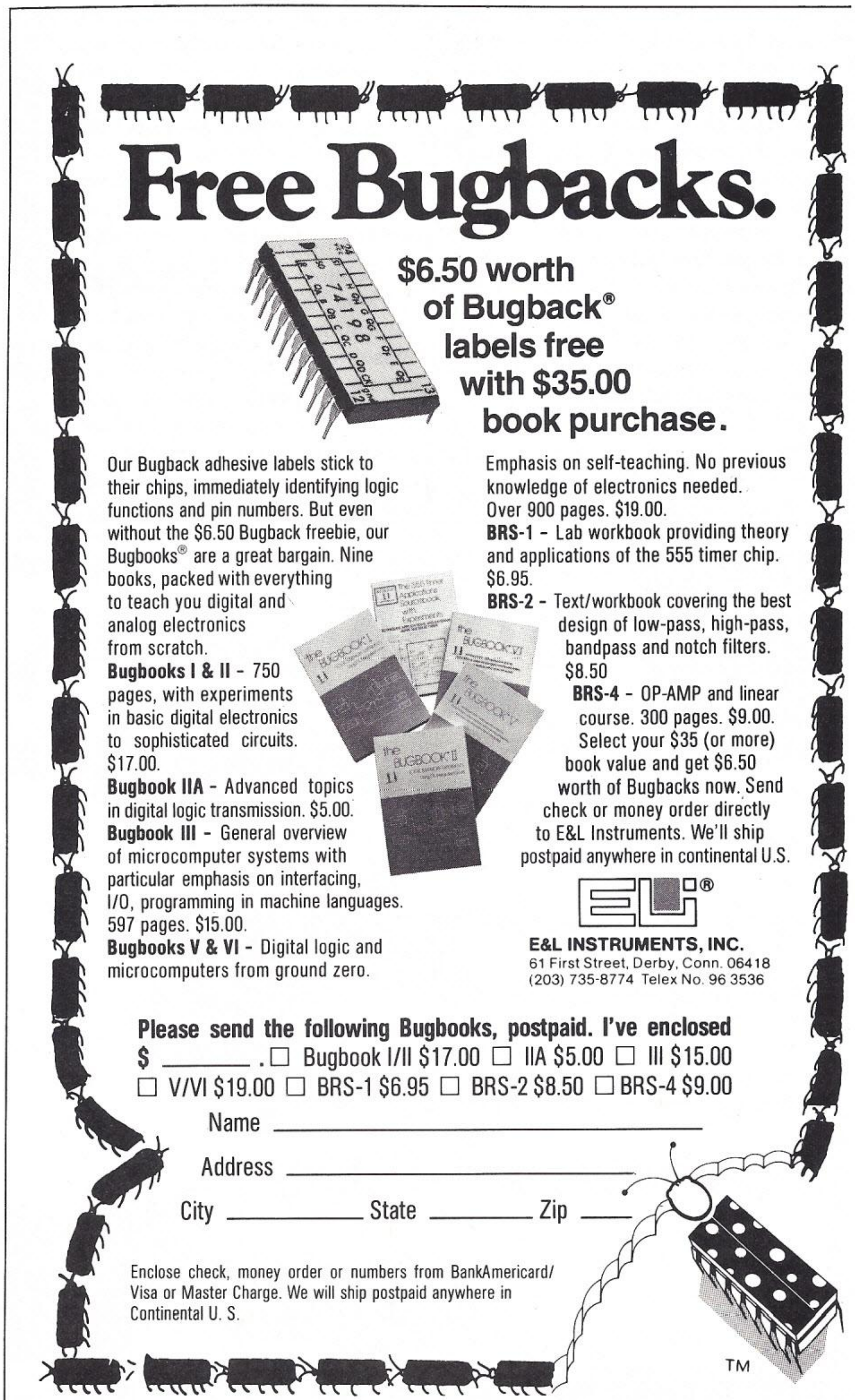
Chauvinist computing

There are three unhealthy currents which underlie many of the articles and attitudes appearing in *Personal Computing.* These are anti-intellectualism, male chauvinism and sexism.

Anti-intellectualism is an absurd stance for a member of the print-media which is trying to capitalize upon the need for an exchange of ideas in a new field. It's like producing a pictureless magazine for illiterates. Male chauvinism is, among other things, a way of managing activities and attitudes without giving people a choice while saying they can do anything they want. Sexism is, in practice, a multitude of procedures which destroy the full human potential of all of us, men or women, though often felt most painfully by women. None of these have a place in personal computing, which is about how people (i.e., women, children *and* men) use small home/business computers.

Let me point to specific data from *PC.*

- Your first issue interviewed an individual at a show who had purchased an IBM 5100. Now, he is one of the elite who has plunked down enough cash to buy a full system. You reported that he had bought the system for games, but that his wife should not be told that.
- In the May/June issue: In the article entitled "Under Cover Computer," the hero is represented as an embattled but enlightened manager who has to hide the use of a small computer in his office from the computing department. The reader is asked to dream up newer and sneakier ways to do this, even to develop a business to solve this problem.
- In the May/June issue: In an article entitled "Dots Break the Ice," the lead-off summary asserts that it is a premise of the Lemonade Computer Service Company (and hence, *PC*) that almost anybody can put off complaints that he is "just fooling around with an expensive toy" by going into business.
- In the May/June issue: Staff designations. The publisher and all editors are men. The head of the art department is a man, and probably production and circulation also. General Administration is the only department headed by a woman which also has a man working in it. So women share as much power and responsibility in *PC* as men? Do they get equal pay for equal work? Did this distribution come about without the exercise of sexist discrimination?
- In the same issue: *PC* advertising on pages 94-95. Notice who is doing what. Balding man doing books — important stuff. Woman doing inventory of groceries — not important to men. Young woman using computer to make yakking easier — not important. Young man doing school work — important stuff. Young man building kits — important stuff. Woman secretary doing word-processing with computer — more dreck. Famous man being interviewed about personal computers. (Why him of all people?) Man wiring-up a CRT — important stuff. Young boy and girl at the end of a game. He won! She makes mandatory infantile gesture. He smirks with everlasting superiority. The strangest thing about this is that the art work had to have been done by one of the women on the staff. One would not expect such blatant sexism in such a case. But I assume the power of approval lay in the male department head.
- In May/June edition: No significant article is by-lined by a woman. Three banal articles are written by Winkless III, none of which took any special



CIRCLE 11

male talent, and any of which might have been better stated from a woman's perspective.

This is data plus commentary: on to interpretation of the data.

• It is downright stupid to imply that the manager should hide his use of a personal computer. He has the income to buy his own computer. His taxes are eating his paycheck alive. He can justify the use of the computer to the IRS as being germane to his work and promotion. He should buy it personally, install it, use it, and then sell it to his superiors. It is intellectually dishonest to dream up subterfuges when a direct approach is valid.

• It is also anti-intellectual to evade the accusation that the computer is an expensive toy by talk of business activity. Many toys are *more* expensive than a personal computer. Boats, recreational vehicles and vans can be more expensive. There is no reason big bucks cannot be allocated to personal computer systems as entertainment, just like every other expensive toy. The real question is interest, hours of use and satisfaction. I refuse to believe that vans, water-ski boats or sailboats are justified by a low cost per hour of use or that they provide equal satisfaction to all members of a family.

• The man who bought his system for games but does not want his wife to know is not only being anti-intellectual, he is being chauvinistic. By not telling his life's partner his real reason for putting up that money, he is denying her the opportunity to participate in the decision. He is also attempting to ratify his exclusive use of the entertainment device in much the same way as men ratify their use of profanity away from women by saying it is wrong to be profane in their presence. There are enough computers for everybody. Personal computing does not need to be reserved for the grubby environs of men without women.

• I cannot help you with your male chauvinism and sexism. The most I can do is shock you into asking the women in your staff and families if there is any truth in the matter. If they do not value and trust you enough to attempt your collective re-education, then you are not worth anyone else's time. One thing that can be said is that men cannot escape from chauvinism and sexism by themselves. Look at it in terms of the higher management in "Undercover Computer." They are not expected to see the light because they are known assholes who, by themselves, cannot be expected to change themselves. The same applies to male chauvinists and sexists. You are basically engaging in asshole behavior. Without the help of people who are outside your asshole perceptions (women, in this case), you have no way out.

Let's go back to basics on the male chauvinism/sexist issue. You have elected not to include cheesecake (female frontal nudes) in your magazine so that it will not be one of *those* kind of publications. You want to be respectable, almost business-like, but not stuffy. *PC* has even complained about the lack of women at the shows, etc. You want to be the leading magazine in the field, to mold attitudes. Yet, the primary attitudes you are propagating are male chauvinism and sexism. No wonder women were not interested in personal computing conventions.

MAILBAG

PC is driving them away in droves. You may as well include cheesecake in your magazine because you are already driving away women with the rest of the chauvinist/sexist presentation. Rather than denying yourself the sales potential of cheesecake, you could balance it with beef cake (male frontal nudes).

It is time for *PC* to shape up or ship out. We need a wink, a pamphlet, and a whole loaf. There has to be more difference between *Field and Stream* and *Personal Computing* than that the former is about the sexual domination of animals and the latter is about sexual domination in personal computing.

Hollis D. Paul
Torrance, California

You're right – *Personal Computing* has contained passages that some people consider anti-intellectual, male-chauvinist, and sexist. But starting with this issue our new expanded staff is editing out most of those passages.

To set the record straight – our art director and production controller are women, and so is one of the new members of our editorial staff. We're eagerly looking for articles by women and men.

More Babel?

Concerning the Hewlett-Packard modified keyboards (May/June '77, p. 130): Why not modify ASCII to cover Spanish, French, Greek, Russian, Arabic and Thai typefaces? Or start a new script form (let's call it "Unity") that, like Cherokee, has one letter per syllable, and allows you to sight read any language even if you can't speak it.

Or conversely, if you hear it you can write it, even if you don't understand what is being said. Perfect for computer secretaries, for teaching a computer to speak or listen, etc.

John Nix
Rockford, IL

Teacher wants data

Fred and Joan Saberhagen's article titled "Homework" in the July/August issue covered many of the things I would like to do with a computer in my classroom. Your article mentioned two topics that I need more information about. Perhaps you have a source that has the information I need.

I would like to be able to use the computer (when and if I get one) to schedule classes. Do you know of any software or other information about using a minicomputer to schedule?

The possibility of grading papers with a computer will work well in my math classes if I can find a reader (at an affordable price) to read the answer sheets. Do you know of any device that would do the job in combination with a minicomputer?

Any information that you have would be of great help to me. Thank you for your time and help.

John Fultz
Manteca, CA 95336

Dear Reader,

Please send us your comments, criticisms, and ideas. Clue us in on where you're at.

Russ, Ellen, Harry

ON YOUR BEHALF

It's in the cards

I read with interest your article "The Programmable Pocket Calculator" in the November/December issue. The article mentioned a "student who built an inexpensive card reader to interface with his HP-25 . . . " I was wondering if somone could provide me with further information on this device. I would be interested in duplicating it for my HP-25.

Melvin L. Zwillenberg
21 Tekening Way
Hamilton Square, N.J. 08690

Doctor seeks medical help

I have been reading your magazine with great interest. I would like to contact any readers who are interested in medical applications of computer technology, particularly information on electrocardiogram interpretation and details on conversion, A/D, D/A, with appropriate interfacing for Z-80 processor; software on any medical item especially diagnosis of endocrine, ovulation disturbances; Medicaid, Medicare, third party billing software; and business software.

Jonathan C. Gibbs, Jr., M.D.
Memorial Health Center
75 Harrison Ave.
Jersey City, N.J. 07304

Convert shorthand

Maybe someone can help me with a problem. I don't have a computer and as far as I can tell the few books I've gotten don't seem to apply to what I need. My town has no school for computing and even the library carries little.

I don't care about doing match problems or playing games. What I'm interested in is programming for word retrieval – sort of. Specifically, converting written machine shorthand back into English.

My shorthand machine is a "computer compatible" system. There is another small machine that attaches to the shorthand machine. As shorthand is taken it is "transferred" to the smaller machine and put on magnetic tape. I don't even know how this is done.

My question is, how would one go about programming for words (or letters – or however it works)? Any ideas where I could find this sort of information?

To buy one of these systems programmed and to rent the machine (or teletype to a computer in another state) could cost $2000 a month! The basic program may take a while to do, but for that much money I'd rather learn how to do it myself.

Susan Gentsch
139 Garden Ct.
Del Rio, Texas 78840

Toys or tools?

I found your November/December "Toys or Tools" article by Jack Roamer very interesting. The article mentions that the Fairchild VCS could be easily modified (electronically and with a little programming). I would like to find an article on how to accomplish this, perhaps one with schematics and instructions. If the software were also to be included, it would be even better. Specifically, I'm interested in modifying the VCS into a microprocessor capable of handling BASIC. I'd appreciate any information you or your readers might have.

William J. Beyda
467 Rose Lane,
Rockville Center, N.Y. 11570

Secret life of plants

A couple of years ago I read a book called *The Secret Life of Plants.* I was very much impressed by the fact that plants produce signals in response to their environment, and I would like to set up a personal computing system to collect such signals from my household plants. I believe that statistical studies of the daily information could offer some insight into the phenomena involved, and help reveal many mysteries of the world around us.

If you know of any other work being done in this area, or of others who have already designed working systems for communication with their plants, I would like to hear from you.

Julia Kalendar
Seattle, Washington

V.I.P. exchange

As a hardware hacker from the old tube days (remember tubes?) gone digital, I've been watching the calculator/computer "revolution" with great relish. Also, as an interstate cheapskate, I have joyfully followed the downward trend in prices due mainly to LSI. My first micro was the MARK-8.

I decided to invest in a new system that promised tape cassette interface, CRT graphics, and the low power I need since moving away from 60-cycle hum. The COSMAC VIP by RCA (1802 system) has enjoyed an absolute dearth of publicity, but I'm now able to store programs on tape using the same low-cost Panasonic that would not work before.

But like all other systems I've looked at, the instructions for using the instruction set are woefully inadequate. The books always seem to be written by someone who already understands the process and they're totally incomprehensible to the novice. Not enough examples are given and those that are given are not explained. Unfortunately, the instructions for a CPU are not what I would choose were I designing the chips and the guys who did design it aren't telling me how to put these instructions together so they will do what I want. Are any of you readers using the COSMAC VIP? If so I would like to exchange notes with you.

D.L. Hartley
Rt. A Box 168D
Yellville, Ark. 72687

Texans, unite!

I greatly enjoy your magazine, but I have a question that you might be able to help me with. Are you aware of any user's group for calculators using the Algebraic Operating System (AOS) of Texas Instrument calculators such as the SR-56 that I own.

I am currently unaware of any such group but am interested in organizing one. If anyone else is interested in joining such a group or knows of an existing one, please contact me.

Terrance A Ward
1408 S. 10th Street #21
Waco, Texas 76706

RANDOM ACCESS

Tale of two cities

Every weekend in Glendale, Colorado, the normal population grows from 3,000 to 75,000 as residents from nearby Denver swarm to the city to take advantage of the city's only industry — entertainment. This weekend population explosion forces the city's police force and city government to deal with 25 times as many people as during the week as well as all the unsavory characters that large crowds and popular entertainment spots attract.

The town decided to buy two IBM System/32s to help both police and city administrators keep things running smoothly, before the weekly overload of visitors could short-circuit the operation of city government and law enforcement.

Glendale uses one of their new computers to handle their water utility billing, payroll, sales tax accounting, and accounting for the city's $3.2 million budget. Their second machine works 24 hours a day to supply the police force with immediate responses to their inquiries.

The police computer system, much to the dismay of local criminals looking to take advantage of the swollen weekend population, acts as a computer vigilante. The machine supplies information on wants, warrants, and past crimes committed in Glendale's jurisdiction (including the type of offense, time of day, location, and arresting officer), descriptions of known felons, fingerprint codes, and methods of criminal operation. The computer also ties into the computer network of the FBI and the Colorado Bureau of Investigation.

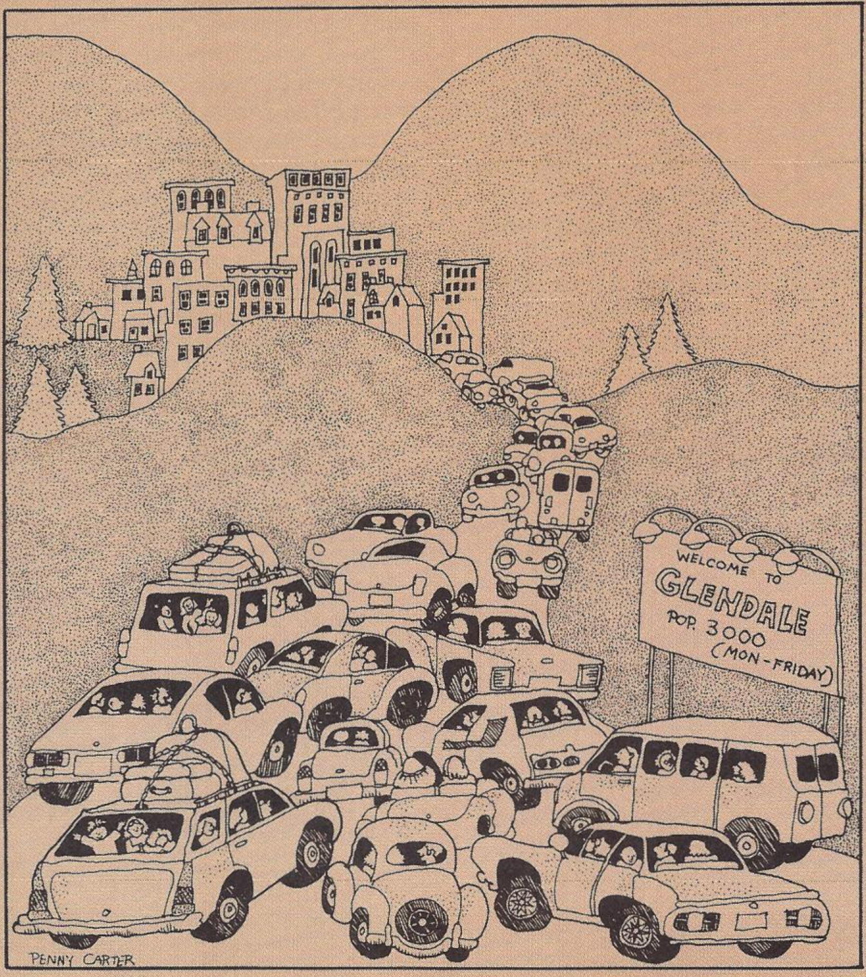
In addition, the police system handles voter registration, liquor card licensing, parking ticket accounting, and the bookkeeping for paid fines.

Satisfied with the two computers, Glendale councilman Tim Greer said they "actually improved the efficient operation of all levels of government" and "eliminated some bureaucratic mistakes which the old, paper-and-pencil techniques permitted."

Because of the computers' present success, Glendale has begun using the machines to automate court records. The system supplies the presiding judge with a complete, computer-prepared printout on each case. The computerized system has already shown signs of speeding up the docketing and hearing processes.

Town residents are also happy with the new computer system — that is until they get nabbed for not paying their parking fines . . .

The Long Beach, California City Hall purchased an IBM computer system to help solve a somewhat different sort of problem. Citizens there complained that bureaucratic red tape had made it next to impossible to register a complaint or make an inquiry at City Hall without first making a dozen calls. And, if they did manage to contact who they were looking for, it was even more difficult to get something done. In response to these complaints from residents, the City Hall employed the IBM/370



model 158 computer, along with an IBM program called the Information Management System, to implement what they call a Citizen Service and Information System. The Citizen Service allows Long Beach residents to dial just one number (590-6555) to reach the proper party at City Hall and be guaranteed an answer.

The service works like this: the caller dials the Information Service number and is connected to a clerk seated at a computer terminal. The clerk takes down the caller's name, address, phone number, and matter of inquiry. The clerk then scans the display screen's computer listing of information areas to decide who should handle the question. Information categories range from animal regulations, bicycle routes, and consumer protection to refuse collection, senior citizen services, and sidewalk repair. Then the clerk connects the caller to the proper party.

After each call, the computer automatically follows-up by printing a letter thanking the caller for the inquiry and giving the caller an inquiry number and department name to refer to should he need to call back. A copy of the inquiry is then forwarded to the district's city council representative so he/she is aware of the district's problems.

If a final disposition on a call isn't received within a predetermined period of time, someone at City Hall calls the department involved to check on the status of the problem.

The computer records all incoming calls and stores the information for future reference and updating. Inquiry updates as well as new ordinance regulations or changes in personnel or personnel responsibilities can be processed immediately and become available to the clerks operating the computer terminals.

According to City Manager John Dever, citizen response has been fantastic. Seven hundred calls were recorded the first two days and about a hundred calls have been coming in every day since.

Statistical analysis of the Long Beach inquiries have pointed out specific areas of public concern, public ignorance about specific laws or situations, and in some cases undetected problems. (For example, a number of reports from the same general area concerning cracked sidewalks may indicate that there is a problem with the roots of trees in the area.) In Glendale, analysis of the computer information has helped pinpoint crime trouble spots and times. Both cities have also used the information to help in planning and formulating new policies and procedures.

So the next time someone tells you how a computer turned off the wrong person's phone, electricity, and gas — or had a warrant out for the police chief's arrest — you can come right back with a story about what's going on in Glendale or Long Beach.

My son, the computer

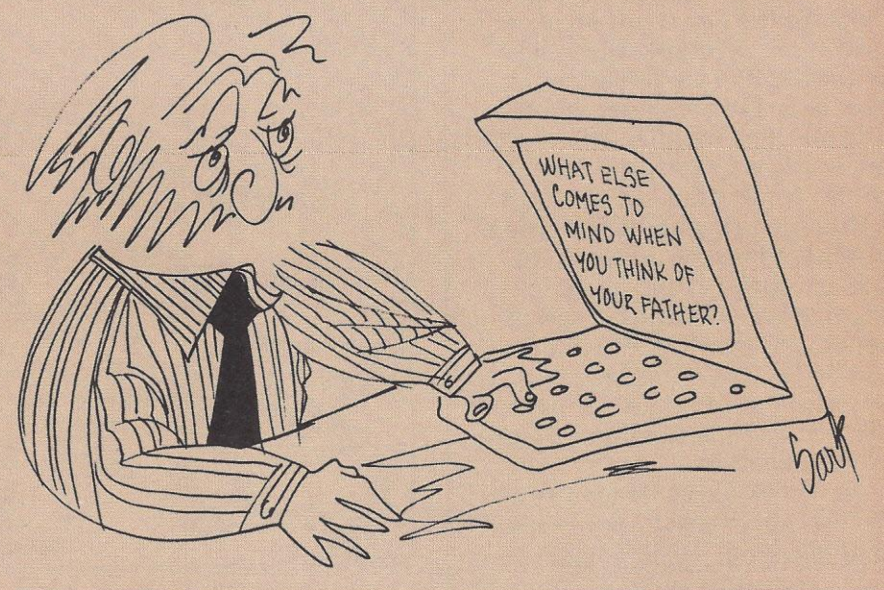
Did your mother always remind you how proud she would be if she could say, "my son (daughter), the doctor?" Well, Mom may soon be having second thoughts about the prestigious profession. The New York University Linguistic String Project (LSP) has successfully applied its programs for natural language processing to medical records. LSP plans to use their programs to process patient data and make "recommendations" to the doctors concerning patient admissions and releases. When fully developed, the computer will also evaluate and analyze doctors' procedures and diagnoses.

Naomi Sager, Lynette Hirschman, Ralph Grishman, and Cynthia Insolio are working on the project at NYU to transform free narrative input into a structured data base suitable for automatic information processing. The processing system will be capable of answering questions, editing records, and compiling statistics.

To establish an appropriate structure for the material, the program's designers first performed a manual linguistic analysis on a sample of the material. From this analysis they drew sets of word classes and an information format. Then, through a series of processing programs, they broke the sentences down into their grammatical structure.

Hirschman, Grishman, and Sager performed the first experiment in automatic information formatting in 1976. Their data included sets of follow-up x-rays from 11 patients who had had breast cancer surgery. They used 159 reports, containing a total of 188 sentential units. 176 of them were formatted successfully.

Among the output for a formatted x-ray entry was a TEST section, which included slots for test time, body location, and date; and a FINDING section, for the verb and information about



change and specific medical findings. For example, if you formatted the sentence "flat place of abdomen — mild degenerative changes and osteoporosis of the sacral spine," the results, which look like a diagrammed sentence from seventh grade English, would be structured like this. Under TEST would appear: plate, flat. Under FINDING would appear: changes, mild degenerative. A second format, "B", includes the categories MED-FINDING (osteoporosis) and PT BODY (spine, the sacral.)

After mapping out the medical narrative into information formats, the data base can be used in a number of ways. Work presently being done processes questions and answers them with information from the data base; this work also extracts medical statistics that will aid in judging re-occuring trends of various diseases.

Another LSP experiment uses the text of hospital discharge summaries to establish a format for processing tasks without having to input the patient record into a special format. Although the formats for these documents involve a greater variety of material, the restrictive and repetitive features of the sublanguage offer the structure necessary to form word subclasses and synatic regularities.

If successful, this process could be applied to automating routine hospital admission screening. The discharge summaries, along with the criteria used to measure if the case merited review, would serve as input. In order for the system to operate, it would also have to include the answers to such questions as: where particular symptoms were noted; where certain procedures were carried out; and whether certain variables were monitored.

The program is still in the infant stage, but if and when implemented in hospitals across the country, all those patients that just visit hospitals because they like the attention (or because they like their doctor — what *would* mother say??) may first have to pass the scrutiny of a computer.

Pay by phone?

In the not so far future we may be able to pay all our monthly bills by making just one phone call. A House Banking Committee's subcommittee on Consumer Affairs is considering legislation to initiate a nationwide electronics fund transfer (EFT) system. The proposed system has already elicited many controversial responses from consumers, banks, and other concerned corporations.

Speaking before the subcommittee, Jane Butel, vice president for consumer affairs for the American Express Company, warned that the sudden initiation of a nationwide EFT system, without previous marketing tests could result in a system that neither fits the needs of, nor pleases, the American public. Butel recommended to the House subcommittee that the implementation of the system be "evolutionary rather than revolutionary."

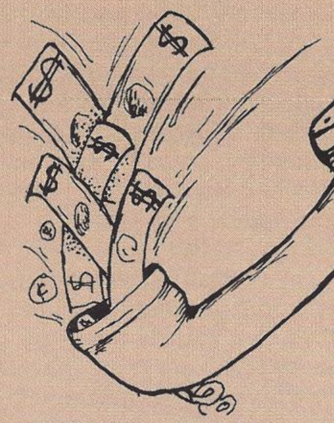
Supporters of the EFT system speak of the money and time it would save — no more postage stamps, personal checks, or having to rely on the mail. The EFT system would also allow you to keep your money in your account until the last minute.

The propsed system structure relies on a 3-part code system to complete a transfer. To "take" money from your account to "pay" your creditor, you would just call your bank, give the operator your account number with that particular store, the store's account number, and your bank account number. The computer does the rest. Since each transfer requires the knowledge of all three code number sequences, the chances of someone else toying with your account are limited.

Yet, regardless of the built-in safeguards of the proposed system, many people are still worried. Speaking for American Express, Butel proposed four additional provisions for consumer protection legislation in her address to the Board.

The first safety measure concerned mail security. Butel recommended that laws prohibit the sending of EFT cards and personal identification codes through the mail to avoid allowing someone easy access to your account due to a lost or stolen envelope.

To ensure account security, Butel also urged using a random method of assigning identification numbers, rather than basing the assignment system on code numbers that relate to already existing numbers such as your drivers license or social security number.



The third proposal concerned requiring an individual (human) to be available during business hours to discuss possible errors, questions, or problems with the system or your account.

The final piece of consumer protection legislation proposed that any national legislation set statutes to insure, as Butel said, "the safety, integrity, and financial security of customers by making sure that EFT systems do not constitute unsound banking practices."

On the state level, some mini transfer fund systems already exist. A Massachusetts savings bank consortium has implemented a "Pay-by-Phone" bill paying service, various airport terminals across the country offer travelers cheque service, and many grocery stores around the country process on-the-spot check cashing.

The final legislation for a nationwide EFT system has yet to be passed. The testimony given before the House subcommittee as well as the results of a minority report done by an earlier National Commission of Electronic Fund Transfers, are still under consideration. So just hang on — that pile of unpaid bills on your desk isn't going to disappear just yet.

Plumber plugs the money drain

Jay E. Moss of Simi Valley, California, knows his business. He owns a plumber contracting firm that does about $2.5 million of large scale contracting annually. A year ago, Moss knew virtually nothing about computers. But because of a magazine article, he now runs his business with the aid of an Altair 8800A microcomputer with peripheral equipment. Moss developed programs for his system to handle the company's payroll and bookkeeping, and to type out invoices and flawless form letters. (Just because Moss automated the payroll, the company's accountant lowered his monthly fee by $75.)

To establish the program by which the computer could compute the contract bid estimates, Moss first fed in data from the records he had kept over the years detailing how long it had taken to do each of 15 operations on a house during construction. With this background information and information concerning the job specifics, the computer can process in 15 minutes (for a tract of 3-5 houses) an itemized job estimate report that includes profit and overhead factors.

To check his computer's accuracy, Moss took the data from old housing plans and fed it through the computer. Based on actual labor and material costs, the results were surprisingly accurate.

Moss initially spent $2,700 for his Altair kit. After adding to his original system a keyboard terminal, large memory boards and floppy disks for greater information-handling ability, his investment has increased to $9000. Moss estimates that most small businesses could put together a functional system for about $5500. You don't have to know a lot about computers, you just have to know your own business.

Home installation?

McGraw-Hill's Personal Finance Letter (PFL) has finally noticed personal computers, and paid the industry a sort of left-handed compliment by suggesting that the value of a new house may be enhanced if it is wired from stem to stern for a home computer system.

Crediting J. David Callan of Pertec for the suggestion, PFL reccommends having the contractor install a 10-wire, low impedance cable and outlet to each major area of your home. This would make the control of remote devices by a central computer easy. Not a bad notion.

The left-handedness of their comment comes in when they say "Don't laugh! Home computers aren't as far out as they may sound." Gee, you mean there are people still out there who don't expect to be equipped with computers as a matter of course?

Surprising isn't it, how easy it is for anyone with even one toe in the water to forget how odd it seems to most people that private citizens may want to take a dip into the personal computing field.

Moving on up there with the big guys

The growing popularity of personal computing has finally made its mark. The Association for Computing Machinery (ACM), a group that now boasts a membership of more than 34,000 computer specialists, announced at the National Computing Conference in June the founding of a new Special Interest Group on Personal Computing (SIGPC). According to ACM, the group will be dedicated to exploring the "educational and scientific purposes in the design and applications of computer systems for personal uses." Included in this study will be the use of personal computer systems in the home, for clerical purposes, in small businesses, for management and recreational purposes, and the applications and techniques of software and hardware technology when integrating graphic, speech, data management, and music systems.

Dr. Portia Isaacson, who chaired the 1977 National Computing Conference, has been named as SIGPC chairperson. Isaacson's plans for SIGPC include holding their first business meeting at ACM '77 in Seattle, appointing other officers, and publishing a quarterly newsletter.

SIGPC membership dues are $5 a year for ACM members, associates, and student members. Membership for non-ACM members is $13 a year. The membership fee includes a subscription to the newsletter, although newsletter subscriptions are also available without membership for $12 a year.

To join SIGPC, write to: the Association for Computing Machinery, P.O. Box 12105, Church St. Station, New York, New York 10249. (ACM members be sure to include your member number.)

Information about other SIGPC programs can be obtained directly from Dr. Isaacson by writing to her at: The Micro Store, 634 South Central Expressway, Richardson, Texas 75080.

Good-bye to tables

Now you can throw away your conversion tables, pencils, and paper while debugging. Texas Instruments has designed a new calculator/converter that lets you make instant calculations for computer programming, operation and study.

The calculator/converter, called the TI Programmer, performs number base conversions and calculations in decimal, hexadecimal (HEX), and octal (OCT); bit-by bit logical operations on numbers in HEX and OCT; and "two's complement" arithmetic also in HEX and OCT bases. The Programmer includes a 1'sC key for one's complement (HEX and OCT) and signed floating point arithmetic for use in day-to-day computation.

For additional work with mini and micro computers, the TI Programmer offers the logical functions AND, OR,, Exclusive OR (XOR), and shift SHF. These fucntions also operate bit by bit on numbers in HEX and OCT.

The handheld calculator has an 8-digit capacity in all bases with a LED display. The 3-key memory lets you store, recall, or sum to memory contents.

The advent of this new calculator/converter took members of the slide rule generation (and TYCON Inc., who had plans to produce a hexidecimal version of their 8080 octal code conversion card) by surprise. Now, with the new Programmer, making conversions and calculations is a simple process. For example: to convert $FE03_{16}$ to Base 10 (DEC) you would just 1) press HEX, 2) enter HEX number FE03, 3) press DEC, and 4) the equivalent 65027 would appear on the screen.

To find the "two's complement" using the Programmer, you would just 1) press HEX, 2) enter 3FE, 3) press =/-, and 4) the "answer" appears on the screen, FFFFFCO2. All the other types of calculations and conversions are just as easy to perform.

At $49.95 the TI Programmer shows promising signs of taking over where the slide rule and Bic Click left off. For more information on the TI Programmer, write to: TI Programmer, P.O. Box 53, Lubbock, Texas 79408.

Space computer makes joyful hams

Until now, amateur radio buffs could only receive and send consistently strong signals from the AMSAT-OSCAR 7 Mode B transponder for about 23 minutes at a time. But according to the American Radio Relay League, plans for 1979 include the launching of a Phase III satellite that will send out the same strength signal over a 12-hour elliptical orbit from about ten times the distance.

The satellite's basic functions will be established on the ground. Once the satellite is off in space, the responsibility for communication with the spacecraft's control system will rest with a CMOS microprocessor to be known as the Integrated Housekeeping Unit (IHU).

If everything goes as planned, the IHU will be able to monitor or control virtually every function of the spacecraft. The system will be programmed to execute telemetry and command requirements as well as make corrective changes in the power and communications system where *it* deems necessary.

In addition, the IHU will establish the time clocks needed for the spacecraft's various and numerous timing functions, adjust the orientation of the spinning body in space, and decide when it's "all systems go" for the kick-motor firing. And it will be the IHU that gives the final command to fire.

Using microprocessors to control satellites is not new. In the past, microprocessors have corrected minor design faults on both the OSCAR 6 and 7 satellites via remote control.

Let's just hope that they work out all the "design faults" before they put a microprocessor in charge of controlling our defense missiles.

Schools get graphics, you get royalties

Computer-based learning materials have been around for a while, but few programs place the major emphasis on computer graphics. That all may change. Educulture, the educational technology division of the college textbook publisher William C. Brown Company, is presently designing a set of computer-based graphically-oriented courseware programs specifically for small stand-alone computers. The secondary and post-secondary educational materials offer coordinated learning series in math, English, and the sciences.

The programs' emphasis on graphics can be attributed largely to the influence of Dr. Alfred Bork, an internationally recognized authority on computer use and a strong proponent of using computer graphics in teaching. Bork, who now holds positions as a physics and information professor, director of the Physics Computer Development Project at the University of California at Irvine and physics chairman of CONDUIT, will serve as a consulting editor on the project.

The program will use pictures, diagrams, and special characters and symbols for math and science along with the written text.

The programs are designed for machines with 32K bytes of random access memory, single drive digital tape or floppy disk storage, and medium-resolution CRT displays.

Educulture's entry into the developing field of computer-based learning is the first major step to be taken by a private publishing firm. Previously, most funds in the field came from government grants, individual schools, and large corporations.

The Brown Company will pay "advances and royalties" to qualified authors — an unusual offering in this field. So, if you're interested in finding out more about writing for Educulture or about their new courseware programs, write to Jon Bosak, Project Editor, Educulture Inc., 3184 "J" Airway Ave., Costa Mesa, California 72626.

Nurses share Plato with mechanics

At a vocational school in Minnesota, two computers alternate between impersonating ailing patients for practical nursing students and imitating chugging cars for industrial mechanics. Designed by the Control Data Corporation in Minneapolis, the PLATO computer-based education system presents its instructional material as text, drawings, and animated graphics on video display terminals.

The computer's program for nursing students includes a demonstration of cardiac functions through animated drawings of valve operations and blood flow. The student can be immediately informed of the particular function of a specific section of the heart by merely touching it on the screen. When the students and PLATO play "doctor," the student, as the medical examiner, evaluates information concerning the "patient's" blood pressure, cardiac function, respiration, and physical appearance. Then, the student prescribes medication and care techniques by typing a diagnosis (a slipped disk perhaps?) onto the keyboard. An immediate response accessing the student's recommendations appears on the screen.

Machinist students at the school follow the same type of format to learn about the operation and maintenance of machinery. Structural designs of lathes, hydraulics, electric systems, and power transmissions are flashed on the screen along with questions. Students use the keyboard or touch-screen method to give an answer, ask questions or ask for help, move the images or "turn the page."

The students at the school feel the computer program has advantages over traditional classroom instruction; most frequently mentioned are the individualized form of self-paced instruction and the immediate feedback (two seconds to evaluate the student's work and inform him of his progress). The students like the private structure allowed by the system; if you don't know an answer, or if the answer you give is incorrect, no one but you and your "teacher" knows.

Vo-Tech also uses the PLATO system for mathematics courses, data processing, and special study programs at the school. The two terminals are linked by telephone lines to a Control Data computer in Arden Hills, Minnesota.

With computers like PLATO acting as patients and other companies working on programs to take over various aspects of the doctor's role (see Random-Access — My Son, The Computer) there's not much room left for us humans in the field of medicine.

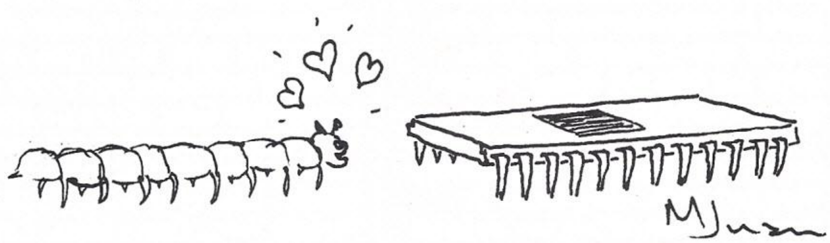
Why do trees sigh?

Wonder no more. The strange noise reportedly heard coming from the forests of California was none other than the trees emitting one collective sigh of relief. Why? Because Fairchild Camera and Instrument Corporation has found a way to leave a few more trees where they stand.

What Fairchild managed to do was reduce its overall paper consumption (including an average of 3.5 million pages of reports a month) by installing a computer-driven IBM 3800 printing system which they hooked into IBM System/370 computers. The new system cuts back on paper usage by printing on 8½ x 11 letter-sized sheets as opposed to the standard 11 x 15 sheets most printers use. The machine puts out 45,000 characters a second or 13,000 lines a minute; printing the same amount of material as a conventional computer-driven printer in one-fifth the time.

Fairchild, a maker of video games, digital wristwatches, electric auto ignitions, and electronic components, uses the IBM 3800 to print the company's accounting data, inventory and production records, and various management reports.

Once other companies follow Fairchild's ecologically-minded move, the next step is to find someone to manufacture recycled 8½ x 11 printing paper.



GIVE A LITTLE...

We'd like to share your story with our readers. So if you, or someone you know, is using a computer to help in an unusual programming technique or application, or just has a humorous story to tell, let us know. We'll follow up on your lead, verify it, and print the story for everyone to read about. Send your story lead to: Personal Computing/Random Access, 1050 Commonwealth Ave., Boston, MA 02215.

BASIC SOFTWARE LIBRARY

Volumes VI & VII are finally released!

Volume VI

Yes and it still contains what was previously advertised. A fully disk interactive business package with A/R, Inv., A/P, ledgers, tax totals, payroll records, more.

As a bonus it also contains the Users Manual for our Firmware Ledger package. These 100 extra pages contain report formats, file creation routines and our very powerful program ACBS1 used to create the powerful file structured data base.

$49.95

Volume VII

Here is that Chess program you have been waiting for as well as a disk interactive Medical Billing package with patient history file.

Also included is our disk interactive Word Processing package (revision Ø).

$39.95

Add $1.50/Vol. for U.P.S. and handling except to APO and PO addresses. Foreign orders add $8/Vol. for air shipment — US dollars only. No purchase orders over $50.

VOLUME I —	$24.95
VOLUME II —	24.95
VOLUME III —	39.95
VOLUME IV —	9.95
VOLUME V —	9.95

OUR SOFTWARE IS COPYRIGHTED AND MAY NOT BE REPRODUCED OR SOLD.

Due to the numerous copyright violations on our earlier volumes — until further notice we are offering a REWARD leading to the arrest and conviction of anyone reproducing our software in ANY way without our written permission. This includes diskettes, paper and magnetic tape, cassettes, records, paper copies, etc.

R. W. BROWN

SRI SCIENTIFIC RESEARCH

220 Knollwood
Key Biscayne, Florida 33149

Phone orders (800) 638-9194

AVAILABLE AT MOST COMPUTER STORES

10% discount on purchases of entire set. Offer expires December 30, 1977

BANKAMERICARD master charge honored

Scaling the Cliffs of Computer Mastery

Part I

BY RAMOND HOWELL

For about a year I heard terms like "bit" spoken with great understanding by some of my colleagues. I tried reading articles about microcomputers, but it was all Greek to me.

Then came my day! Something clicked! The pieces began to fall in place. What was the key that unlocked the confusion of terms and brought order out of chaos for me? Will the same key work for other floundering novices?

There is nothing magic about computers. The first computer was built by a joint effort in the 1930's of Harvard University and IBM – using relays. Now what could be so confusing about a relay? It is either open or closed. True, there was a bunch of relays in that machine, but it was just plain digital logic.

Now I've made my second assumption – you know something about digital logic. Don't let that throw you, though. Digital logic just means that computers operate using on-off states of relays, tubes or transistors. More on that later.

Simple computer

Using Figure 1 we will begin to look inside of a computer. The heart of a microcomputer is called the central processing unit or CPU. This is the part of the computer that does all the work of data processing, arithmetic, moving data, receiving and interpreting instructions and all the other mysterious things that make computers seem superhuman.

Another item that makes a computer possible is Memory or a place to store data. From Fig 1 we see that the arrow connecting the CPU and Memory points in both directions. That means the CPU can store data in Memory and also retrieve data from Memory. Memory is made up of many locations, each being a place to store data. Each location has a specific address. More on that later, though.

A computer is designed to respond to a set of instructions, which are used to program a computer to perform a given task. Some instructions are made up of two parts: the operator and operand. The operator is the basic instruction such as LOAD. See Fig 2. The operand is what the operator acts upon. For example, "LOAD" is the operator. "What is loaded" is the operand. Other instructions are complete in themselves and contain only an operator. An example of this is the instruction CLEAR. The best way to understand how instructions are used is by an example.

One more fact before we look at a sample program. We said Memory is for storing data. Memory is also for storing programs. The program is stored (we will see later how the program got there) in consecutive Memory locations and the CPU starts at the first location and executes the program by performing the instruction at that location and moving on to the next location for the next instruction.

Sample program

Now let's look at the program in Fig 3, which uses the instruction set in Fig 2. This program will solve the equation A + B = C. Our program in Fig 3 begins at Memory location #1 with the operator CLEAR. This instruction is complete in itself and tells the CPU to set itself to zero.

When the instruction is completed, the CPU moves to the next Memory location and gets the next operator, which is LOAD. From Fig 2 we see that the instruction LOAD does have an operand, which is in Location #3. Thus, the LOAD instruction occupies two Memory locations. When the CPU gets the LOAD operator, the CPU looks at the next Memory location to find *what* number is to be loaded into the CPU. Memory location #3 contains the number 4; so when the LOAD instruction finishes, the CPU will contain the number 4.

The CPU gets its next instruction from the next Memory location. The instruction is ADD. From Fig 2 we see that the ADD instruction causes the CPU to get the contents of another Memory location and add that to the number that is already in the CPU. The CPU, by the nature of its design, knows to look at the next Memory location to find *where* the number is that it is supposed to add to the contents itself. The CPU has been containing the number 4. Memory location #5 contains the number 11, so the CPU will add the contents of Memory location #11 (which is the number 3) to the number 4, so the CPU contains the number 7. At the completion of the ADD instruction in location #4, the CPU will be ready to look at location #6. If you look back over what the CPU has done since it started at location #1, you will see it has added the number 4 and number 3, so that the CPU contains the number 7.

Now for the next instruction, in Memory location #6, which is STORE. Look at the instruction set in Fig 2, and see if you can determine what action the CPU will take before reading further. Now let's see how well you did. The operator STORE tells the CPU to look at its operand in the next location and see where the data in the CPU is to be stored. The next location contains the number 12, so the CPU will store its contents (the number 7) in Memory location #12. Is that what you had reasoned out? Congratulations!

The last instruction in our program in Fig 3 is HALT. This instruction is complete in itself and has no operand. When the CPU receives this operator, it will cease operation until it receives further instructions from the outside world.

What is left in the CPU, after it has stored its contents in Memory location #12? It still has the number 7. That's why the first instruction in a program should be CLEAR.

Note the difference between the operand in LOAD and the operand in ADD. See Fig 2. In LOAD, the CPU will load itself with the contents of the operand while in ADD the operand points to the location where the data is stored.

The instruction set we have used in our sample computer has only seven instructions. Though our sample is simpli-


014827
ANNIE GUSMAN

FIGURE 1 – INSIDE THE COMPUTER

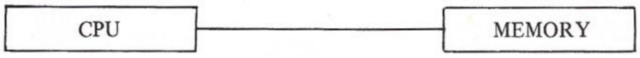


FIGURE 2 – INSTRUCTION SET

OPERATOR	DOES OPERATOR HAVE OPERAND?	EXPLANATION OF INSTRUCTION
CLEAR	No	Sets CPU to zero
LOAD	Yes	Loads CPU with data in operand
STORE	Yes	Stores Data in CPU into memory location specified by operand
RETRIEVE	Yes	CPU gets data from memory location specified by operand
ADD	Yes	Adds contents of memory location pointed to by operand to contents of CPU and leaves results in CPU
SUB	Yes	Same as add except it subtracts
HALT	No	Stops CPU action

FIGURE 3
SAMPLE PROGRAM

MEMORY LOCATION	INSTRUCTION
1	CLEAR
2	LOAD
3	4
4	ADD
5	11
6	STORE
7	12
8	HALT
9	
10	
11	3
12	

FIGURE 4
BINARY VERSUS DECIMAL

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

FIGURE 5 – ASCII CODE

BINARY	CHARACTER	BINARY	CHARACTER
00110000	0	01001000	H
00110001	1	01001001	I
00110010	2	01001010	J
00110011	3	01001011	K
00110100	4	01001100	L
00110101	5	01001101	M
00110110	6	01001110	N
00110111	7	01001111	O
00111000	8	01010000	P
00111001	9	01010001	Q
00111010	:	01010010	R
00111011	;	01010011	S
00111100	<	01010100	T
00111101	=	01010101	U
00111110	>	01010110	V
00111111	?	01010111	W
01000000	@	01011000	X
01000001	A	01011001	Y
01000010	B	01011010	Z
01000011	C	01011011	[
01000100	D	01011100	]
01000101	E	01011101	\
01000110	F	01011110	↑
01000111	G	01011111	←

fied, some microcomputers have as few as 50. But involved and complicated processes can be performed with these instructions.

Binary number system

By now you must be wondering how that thing called Memory works. Before we can understand Memory, we must learn about binary. Binary is a system of counting in which only two numbers are used, 1 and 0 (zero). In our good old decimal number system when we count "... 7, 8, 9, ... " the next higher number is made by using the lowest digit (the number 1) in the ten's place and starting over again in the "ones" place. Thus we have "... 8, 9, 10, 11 ... 19, 20, 21, ... " etc. The same is true of the binary number system. Examine the example in Fig 4.

We find it takes four binary digits to express our decimal number eight. Carrying this further, five binary digits are needed to write Decimal 16, and seven binary digits are needed to express Decimal 64. Cumbersome? Yes, but you see, computers can only work with 1's and 0's, or on-off states.

Lest you think a binary number system is not useful, consider Fig 5. The ASCII code is used on Teletype machines. The code is also a standard for microcomputers and many other data-handling systems. Looking at Fig 5, we see that each ASCII character is made up of seven binary digits, each digit being a 1 or a 0. Each binary digit is called a "bit." Each ASCII character is composed of seven bits plus an eighth bit that can be used as a parity bit. (Parity bits are used to check accuracy of transmitted data). These eight bits form one "byte" of data. Thus a "byte" is eight "bits" of data that are treated as one character, or letter, or number, etc.

Now back to our memory. Each Memory location is made up of eight bits, which compose one byte. The computer sees each byte of data as an eight-digit binary number. Each instruction in the program (which is stored in Memory) is an eight-digit binary number as far as the computer is concerned. Everything the computer does, it does using eight-digit binary numbers.

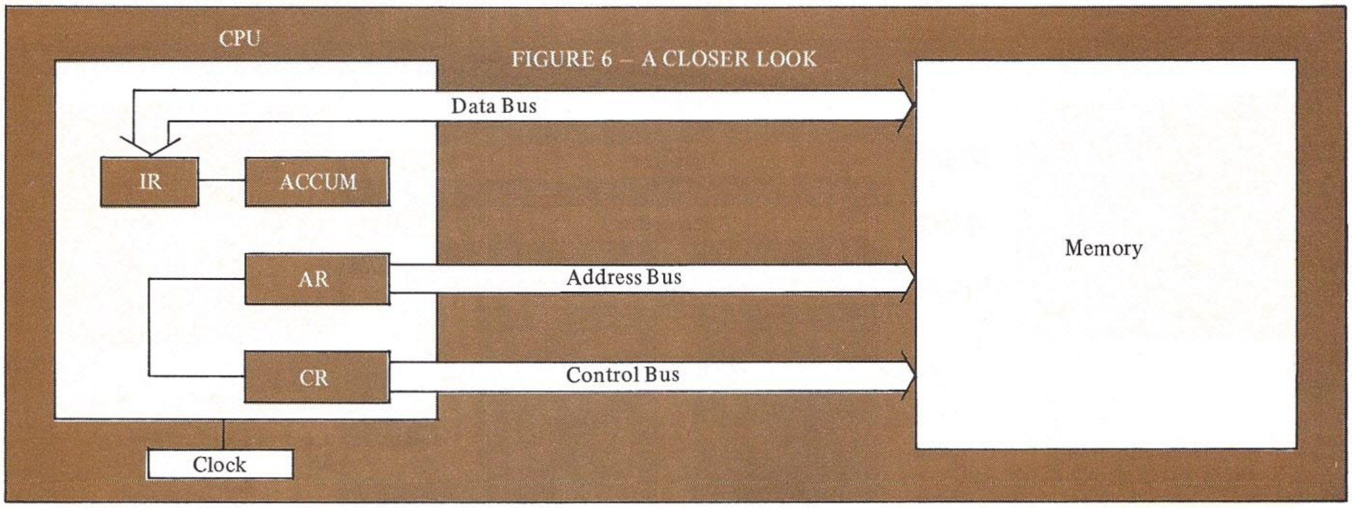
CPU and buses

We have talked a lot about the CPU in a microcomputer. Surely there must be more than what we've seen. Yes, there is – and Fig 6 will help us understand it.

We still have the two main blocks, the CPU and the Memory. Notice the three arrows connecting the CPU and the Memory.

The proper term for these arrows is "bus." A bus is really a set of parallel conductors that convey data from one place to another. The DATA bus passes data from the CPU to the memory. Recall from our discussion of a program that the CPU receives data from a Memory location and also stores data in a memory location. This data was passed back and forth over the DATA bus. In a typical microcomputer, the Data bus consists of eight parallel conductors, one conductor for each bit of data. Thus one byte of data is handled as eight parallel bits of data, each bit handled simultaneously.

The data bus is connected to the CPU and to *every* location in the Memory. That is, when the CPU puts data on the Data bus and wants to store it in Memory location #12, that data shows up at every Memory location. How does



Memory location #12 know it is to receive that data and all other locations know not to accept that Data? That is the task of the Address Register (AR) in the CPU.

There is another arrow, labeled ADDRESS. This is the Address bus. Recall that a "bus" is a group of parallel conductors that conveys data from one place to another. A typical microcomputer has 16 parallel conductors in its Address bus. The Address bus like our Data bus is connected to every Memory location in our Memory.

How does Memory location #12 know to receive the data that the CPU has put on the Data bus? At the same time the CPU put the data on the Data bus, the CPU through the use of the AR put the address for location #12 on the Address bus. Thus only Memory location #12 recognized there was data on the Data bus. All other Memory locations just sit there, content with the data they already contained.

Expanded sample program

With an understanding of the Data and Address buses, and how a given Memory location can be addressed, let's go back to our sample program of Fig 3 and see how our expanded microcomputer of Fig 6 handles the instructions.

The first thing is that the Address Register (AR) would have to be set to Memory location #1. Location #1 is addressed via the Address bus so that the data in location #1 is now put on the Data bus, the Instruction Register (IR) in Fig 6 looks at the Data bus, takes the data that is on it, interprets it, and sends necessary instructions to the rest of the CPU.

The instruction that the IR has just received from location #1 is CLEAR. From our instruction set in Fig 2, we see that CLEAR instruction tells the CPU to set itself to zero. The part of the CPU that is set to zero is the Accumulator (Accum). So when the IR receives the CLEAR instruction, it sets the Accumulator to zero and then tells the AR to move to the next address. The CLEAR instruction has now been completed by the CPU and it is ready for the next instruction.

The AR is now addressing Memory location #2 and the instruction is LOAD. The data in location #2 is now on the Data bus. The IR gets the instruction from the Data bus and interprets it. The IR knows the next action is to look at the next Memory location to find what to load. So the IR tells the AR to address the next Memory location, which is location #3. Now with the data in location #3 on the Data bus, the IR reads that data and loads it into the Accumulator. The microcomputer will continue to move on thru the program using its IR, AR, Accumulator, Data and Address buses and Memory locations as described above, until it reaches the HALT instruction in Memory location #8.

If you have followed and understood this article so far, I am sure you have many questions. The question we will discuss first is "How does the CPU know when to do what?" The word "when" in this question brings up the matter of *time.*

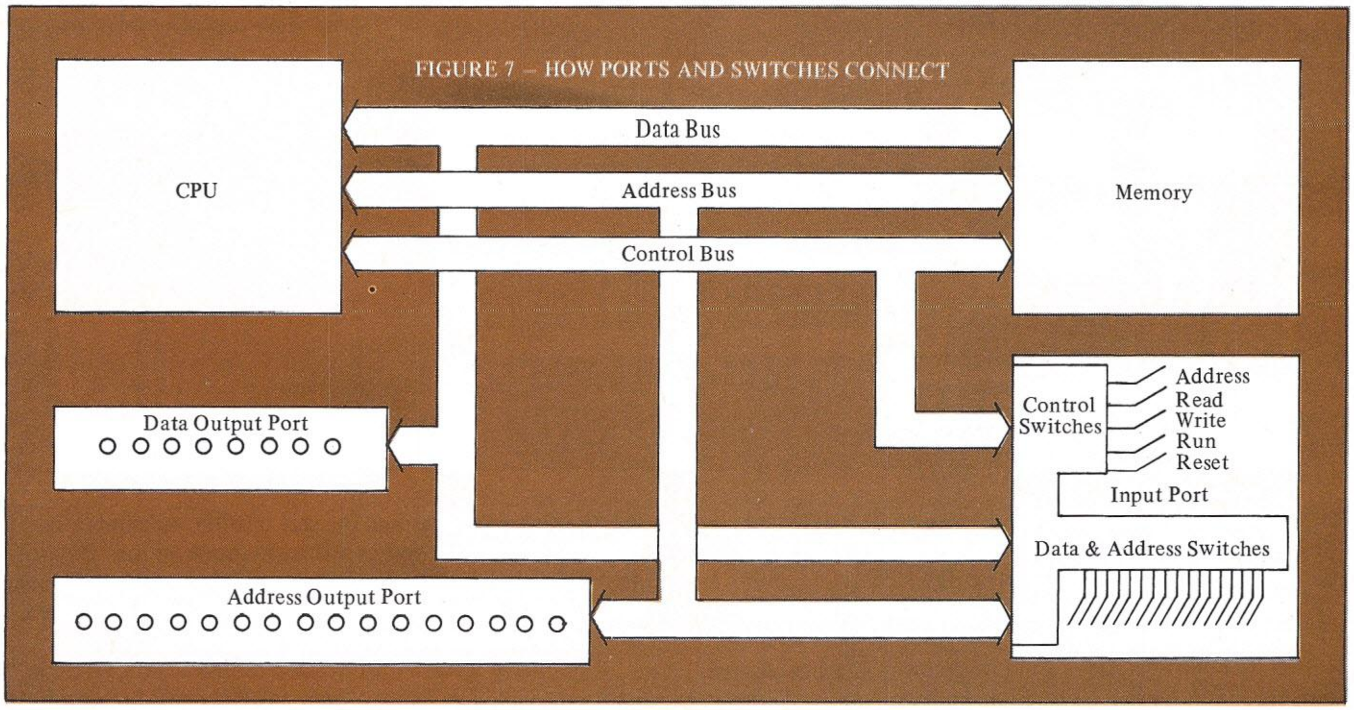
Timing and control bus

Timing is a critical part of the microcomputer. The Accumulator must know when to put its data on the Data bus. The proper Memory location must know when to store data that is on the Data bus. When does the IR receive an instruction from the Data bus, and when does it tell the Accumulator to put its contents on the Data bus? There are a whole host of events going on in a CPU, and the device that sets the pace for all of this is a CLOCK.

The Clock in a microcomputer is nothing more than a special oscillator or multivibrator that produces a signal that steps the CPU through its paces. Tied in with the CPU clock is another bus (called a CONTROL bus) and another register in the CPU (called a CONTROL REGISTER, CR).

The CONTROL bus carries several kinds of information necessary for the microcomputer to operate properly. The only one we're concerned with now is the READ/WRITE (R/W) control line. READ and WRITE are two new words for you to add to your growing microcomputer vocabulary. The word READ means the CPU receives the data from the Data bus, that is, the CPU Reads the data that has come from a Memory location and puts it in either the Accumulator or the IR. WRITE means the Accumulator puts its contents onto the Data bus. In other words, the Accumulator writes its contents into the Memory via the Data bus. The Control bus, which includes the R/W line, goes to *every* memory location, just like the Data and Address buses. When a Read signal is put on the R/W line, all memory locations receive that signal, but only the location that is addressed will respond.

For the IR to read an instruction from the Data bus, several things have to happen in the proper order. This is where the CPU clock and Control bus fit in. First the AR addresses the desired location, let's say location #6 in Fig 3.



The CR then instructs Memory location #6 via the R/W line in the Control bus to read, thus the contents of location #6 are put on the Data bus. Next the IR will read the data from the Data bus, interpret it and proceed from there. The data that the IR read from location #6 was a STORE instruction, so the CLOCK will step the CPU along, the AR will address the next memory location, the CR will send out a Read signal on the Control bus causing location #7 to put its data on the Data bus. The IR reads the data and now knows where to store or write the contents of the Accumulator.

The clock continues to step events along, and the next action will be the Accumulator writing its contents onto the Data bus: the AR will address memory location #12 (location #12 was pointed to by the operand contained in location #7), and then the CR will send out a Write signal on the Control bus, causing the data to be written into Memory location #12.

You've come a long way. Now you know the difference between a "bit" and a "byte," that data can travel in two directions on the Data bus, that a memory location must be addressed via the Address bus and also be instructed via the Control bus to Write before data on the Data bus can be written into it. You've learned that a Program is a useful routine that uses operators and operands from an instruction set, and that a program is stored in memory where the CPU can execute it, by starting at the beginning of the program and performing each instruction as it comes to it, and that things are kept moving by the Clock. You've learned well. Congratulations!

Input/output ports

It's about time we found a way to talk to our microcomputer. It would also be desirable to hear what the computer has to say. This is a task for "Input/Output ports," more commonly called "I/O ports." Anytime the microcomputer deals with the outside world, it does so thru an I/O port.

First, let's look at an output port. One way to know what the CPU is doing is to connect eight lamps to the Data bus, one lamp connected to each of the eight data lines. This way we can monitor each bit of the byte of data. See Fig 7. Recall our discussion of the binary number system? By considering each lamp that is burning to be a 1 and each unlighted lamp to be a zero, we can know what information is on the Data bus.

A set of 16 lamps connected in the same way to the Address bus will also tell us what memory location is being addressed. Now we have two output ports that let us know what the CPU is doing.

Talking to the CPU is more involved. The input port in Fig 7 has eight Data Switches that are connected to the Data bus. The Data input port appears to the CPU as a memory location and therefore has the Address and Control buses connected to it. Since the input port appears to the CPU as a memory location, the microcomputer must contain a program designed to read the Data input port and perform the instruction that is put on the Control bus by the Control Switches in the input port. Let's call this program the Input Program. There are three ways to get a computer into this Input Program. The first way is to just turn the power on. Another way is to press the Reset switch in the Input port. The third way is for the CPU to reach the instruction HALT.

The I/O ports we've discussed are mounted on the front panel of our microcomputer, where they are well labeled and convenient to use. See Fig 8 for a typical layout.

Loading via the input port

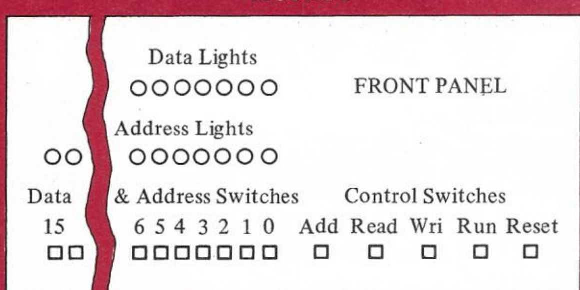
To understand how the Input port of the front panel can perform a useful task, we will load the sample program of Fig 3 into the memory. Since we can only use binary, we must have binary numbers for the operators of the instruction set. In Fig 4, you'll find the binary equivalent of each decimal number used in the program.

The first step in loading the program is to Reset the CPU by pressing the Control Switch marked Reset in Fig 8. Now set up the Data Switches with the address of the first memory location, since the instruction is to be stored at location #1. Location #1 is represented by 00000000 00000001 in binary, so all switches will be off (representing 0's) except the rightmost switch (which will be representing a 1). Pressing the Control Switch marked Address will cause the CPU to address memory location #1. We can examine the front panel Address lamps and verify that the desired address is on the Address bus.

The information we want to put in location #1 is the code for the instruction CLEAR, since it is the first instruction in our program. From Fig 8, we find the code for CLEAR is 01101100. We set switches 7 through 0 (on the front panel) to correspond to the binary code for CLEAR and then press the Control switch marked Write; the CPU will write the data on the switches into the memory location shown by the Address lamps. The data output lamps will verify that the data was handled. We have now stored the operator of the instruction CLEAR in location #1.

Since CLEAR is an instruction complete in itself, it has no operand, so we are ready to load the next instruction. We set the switches to 00000000 00000010 (which is binary for decimal 2) on the front panel and press the Control Switch marked Address. That calls up memory location #2, where we will store the next instruction. Next we put 01101010 (the binary code for the LOAD instruction) on switches 7 through 0 and press the Control switch Write. This will store the code for the LOAD instruction in memory location #2. The operand of the LOAD instruction is the decimal number 4, which is 00000100 in binary. So we call up memor location #3 by use of the switches and press the Address control switch as we did before. Then we put 00000100 (binary for decimal 4) on the Data Switches and press the Control Switch marked Write, so the third step in our program is loaded into the memory of our microcomputer.

FIGURE 8



CODED INSTRUCTION SET

OPERATOR	OPERATOR CODE
CLEAR	01101100
LOAD	01101010
STORE	01100101
RETRIEVE	01101001
ADD	01101000
SUB	01100011
HALT	01100110

We continue on this way until the entire program is loaded into the memory. To make the CPU execute the program, we put 00000000 00000001, the beginning memory location of the program, on the switches and press the Address control switch, which causes the CPU to set the AR to that location. With the CPU pointing to the beginning of the program, we press the Control Switch marked Run, and execution of the program will commence and continue until the HALT instruction is reached or the Reset control switch is pressed.

As the program is being executed, the Data and Address lights will be flashing, to indicate that the CPU is stepping thru the program. When the lights stop flashing, the CPU has reached the HALT instruction and stopped. Our program in Fig 3 was to store the final result in memory location #12. To see what that result was, we need to set the switches to 00000000 00001100 (binary for memory location #12) and press the Address control switch. Now press the Read switch and the Data lights will display the contents of memory location #12. Of course the Data lights will give the results in binary, and we will have to convert to decimal for the results to mean much to us.

Conclusion

We have been inside our sample microcomputer and found our way around without too much difficulty. Now you know several things about microcomputers:

- A microcomputer is made up of a CPU, memory, and I/O ports. Buses connect those three together.
- The CPU contains an Instruction Register, Accumulator, Address Register, and Control Register. The CPU receives instructions, moves data, does arithmetic, and controls memory and I/O ports.
- Instructions put together in a proper order compose a program, which is stored in memory where it can be executed by the CPU.
- Memory stores data as eight binary digits, called bits. Eight bits of data is one byte of data.
- Buses convey data, addresses and control signals between various parts of the microcomputer.
- Input-Output ports communicate between the microcomputer and the outside world.

I hope by now you've gained enough insight into the basic concepts of computers that you can begin to appreciate some of the tougher magazine articles, purchase a computer of your own, and understand the operation manual that comes with it.

At the beginning of this article I asked, what was the key to unlock the mysteries of microcomputers for me? This article is not the key but will lead you to the keyhole. The key? It is simply "hands on" experience. You should have lots of questions that this article has raised and not answered. Only as you get your hands on the microcomputers will you be able to prove that a program will run, or that it will not run. The fact that you have proved it one way or the other means that you've learned something and you will be able to build on the experience.

As you begin to work with an actual microcomputer, refer back to this article. In the next issue, I'll help you bridge the gap from where you are now to the point where you'll be able to find answers to your questions in technical manuals and magazines.

Magic for Educators

MICROCOMPUTERS

BY LUDWIG BRAUN

Personal computers are computers small and inexpensive enough so an individual can afford them and use them for a variety of personal purposes, such as education, entertainment, and management of personal finances. They give teachers, private individuals, and especially youngsters learning environments that were undreamed of only a few years ago.

History of educational computing

Until 1965, few educators used computers in their educational activities, because:

- Few educators knew anything about computing; fewer had thought about bringing this tool to bear on learning.
- The cost of computing was too high.
- Most computers were operated in a batch-job environment. This focus on maximizing machine usage made inefficient use of human intellectual power. (Imagine blowing into a clarinet or plucking a banjo string and waiting till tomorrow to hear the note you played!)

About 1965 Corbato at M.I.T. and Kemeny and Kurtz at Dartmouth came to the rescue with timesharing systems,

This report was prepared at the request of the National Institute of Education, but any opinions are those of the author and don't necessarily reflect the policy of opinions of the National Institute of Education or its staff.

Illustration by Nancy Lawrence

which enabled individuals to work interactively with computers (an enormously-important pedagogic advantage over batch processing), while the computers still worked very efficiently. Computing still was expensive, and was confined largely to universities, but at about this same time the Pierce Report proposed an increased focus on the development of instructional applications of computers in education. In response, the Federal Government (through the National Science Foundation and the Office of Education) has invested many millions of dollars in major national projects:

- PLATO has developed impressive hardware and software and an extensive body of courseware. (The term *software* here refers to the system software, such as the operating system and compilers, while *courseware* refers to user programs employed in instruction.)
- TICCIT has developed an interesting hardware delivery system and a powerful courseware development system.
- Projects Solo and Soloworks have developed novel ways to expose high-school students to computers, and substantial amounts of mathematics courseware.
- The Huntington Computer Project has identified a variety of ways to use computers in high-school science, mathematics, and social-science classes, including many simulations in biology, physics, and the social sciences.
- The Physics Computer Development Project has demonstrated the effectiveness of interactive computation and graphic displays in undergraduate physics courses.
- MATHLAB makes the computer help the student do differentiation, polynomial factorization, and Laplace transforms.
- LOGO's "turtle" teaches second-grade children to develop very complex algorithms for controlling the turtle's motion.
- CRICISAM, a calculus course, includes a computer that helps the student learn difficult concepts and develop the algorithmic approach to problem solving.
- The CASE system and the Lister Hill Network let medical students diagnose and "treat" the computer, which simulates a patient and presents a wide variety of symptom-disease complexes.
- IMPRESS lets students access a large social-data base and extract a variety of statistical reports from the machine, without the usual difficulties of massaging data bases.

Out of all these efforts, we can distill two major classes of educational use of computers: teaching *about* the computer; and teaching *using* the computer as a tool (or medium, or environment). In more detail, the categories of use are computer programming, computer science, computer literacy, computer-assisted instruction (CAI), computer-managed instruction (CMI), and computer-augmented learning (including problem solving, development of algorithms, simulations and games, reorganization and retrieval of information, and mathematics laboratory).

At the same time, people at Digital Equipment Corporation (such as Dave Ahl and Eli Glazer) and Hewlett-Packard (such as Jean Danver) committed themselves to addressing the computing needs of educators. The DEC and HP commitment was aimed at the low-budget pre-college educator who couldn't afford the steep charges required by the time-sharing systems operated by universities and service bureaus. Only because of the price breakthroughs of DEC and HP did the innovative demonstrations mentioned earlier become transferable to the classroom.

DEC and HP made available stand-alone single-user minicomputers and mini time-sharing systems at prices low enough that it no longer was necessary to keep the computer running full time in order to justify the cost (though when such machines were installed in schools, students usually did keep them busy full time). By 1975, it was possible to buy these minicomputers at about $5,000 per terminal.

In the January 1975 issue of Popular Electronics, the lead article described a computer (the Altair 8800) which you could buy in kit form for only $339. This article was the opening shot in what has been a revolution in computer hardware – the advent of the complete, low-cost microcomputer. In the two years since that article was published, over 25,000 microcomputers have been sold, dozens of companies have entered the market place with new computers, terminals, disc drives, speech synthesizers, D/A and A/D converters, audio-cassette-recorder interfaces, and graphics systems.

By early 1977, it was possible to assemble a quite powerful computer for less than $2,000 total, including a terminal and a mass memory device. This machine uses an excellent BASIC interpreter. Such computers now are quite reliable and execute programs rather quickly (up to five times as fast as the PDP-8). That the revolution is upon us is indicated clearly by the new publications, stores, clubs and shows.

Publications

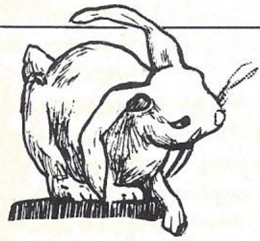
Before 1975, except for an occasional feature article in *Popular Electronics, Radio Electronics* or *73,* one had to go to the professional literature to read about microcomputers. This professional literature focused on industrial applications of micro*processors* rather than their integration into systems called micro*computers*, and was incomprehensible to the educator and the amateur.

Since 1975 new high-quality journals have arisen, including *Byte, Dr. Dobb's Journal, Kilobaud, Personal Computing* and *ROM.* All aim at the computer amateur and the educator. In addition, such "old-time" journals as *Creative Computing* (circa 1974) and *People's Computer Company* (circa 1973) have devoted many pages to microcomputers and their educational and hobby applications.

Computer stores

Before October, 1975, if you wanted to buy a computer, you went directly to the manufacturer, to a manufacturer's representative (usually just a sales office), or to a company that sold surplus computer hardware. The assistance which you could expect from any of these sources was (and is) minimal or totally non-existent, because of the very low ratio of salespeople to customers.

In October 1975, Lois and Richard Heiser opened The Computer Store in Santa Monica, California – a retail store set up much like any other local specialty shop. Potential customers are encouraged to drop in, try out the demonstration units, swap yarns, seek help with a hardware or software problem, attend classes on various computer related topics – and to buy computers (assembled or in kit form), peripherals, components, supplies, and publications. In addition, customers may bring their ailing computers to the Heisers for servicing. When this store opened, few thought it could succeed, but it has succeeded to the point where it has more than doubled its staff and has expanded into much larger quarters. More important, though, than the



success of the Heisers is the proliferation of such stores across the country. There now are hundreds of such stores. They are in or near most major cities in the United States.

Computer clubs

Before the first announcement of the Altair computer in January 1975, there were few computer clubs. Those which did exist were in high schools with computer access, and were concerned only with writing programs. Their memberships were small.

Since then, hundreds of computer clubs have popped up across the country, and broadened their focus to include construction of computers, design of peripherals and interfaces, and unusual applications (ham radios, weather stations, etc.). The most spectacular of these new clubs is the Southern California Computer Club, with a membership over 20,000 drawn from computer enthusiasts nationwide.

Personal computer shows

The national level of interest in microcomputers is obvious from the number of computer shows aimed at the microcomputer user, with focus on the amateur. In the Spring of 1976 MITS (manufacturer of the Altair 8800 and 680 computers, and the IBM – or DEC – of the microcomputer business) held its first annual national Altair meeting; thousands attended. In August 1976, the first Personal Computer Conference was held in Atlantic City with an attendance of 5,000. *Personal Computing* magazine scheduled three regional personal computer conferences for 1977. The First West Coast Computer Faire was held in San Francisco in April 1977 and had a strong focus on educational applications.

Are kits the answer?

In September 1975, HummRo held a conference to explore the computer needs of educators and extrapolate current research trends one decade into the future. The conference concluded that, well before 1985, we'll see "zero-cost" computers, i.e., computer hardware that has negligible cost. This cost trend has existed for several years in the calculator market and more recently in the electronic-watch market; and there is every reason to believe that microcomputer costs will follow suit as the annual sales volume increases.

As exciting and portentous as these developments were for education, one major problem still inhibited the use of these machines. The microcomputers offered for sale before the end of 1976 were intended for the engineer and the hobbyist (indeed, they were called *hobby* computers). Most such computers were sold in kit form and required substantial skill in assembly and in testing to ensure proper performance.

But most educators are not engineers or hobbyists, and don't operate at the needed technological level. And they aren't prepared to accept delayed deliveries, inadequate performance, unfulfilled promises, and all the other difficulties that hobbyists have accepted as part of the price of entering this exciting new field.

To use an analogy . . . the educator who wishes to show a movie wants to purchase a film projector which need only be taken out of the box and plugged into an outlet. He will reject using film if he must first assemble the projector from its parts, or if he receives the projector complete except for the lens and the line cord. The hobbyist, on the other hand, is willing to scrounge up an old line cord from somewhere and build a jury-rigged lens system so that he can use the projector.

Fortunately, in April 1977 at the First West Coast Computer Faire the Commodore Business Machines Company (one of the leaders in marketing inexpensive pocket calculators) announced a computer that costs only $595, comes completely assembled, and is designed for the non-technical home and school environment. During the summer of 1977, Radio Shack began selling a similar computer.

What are the inhibitions?

Despite all the efforts described above, the computer has not penetrated deeply into the educational system. Experts estimate that 25% (about 8,000) of all highs chools in the United States use a computer for educational purposes; but since most of those schools have only one or two terminals, the computer's impact on most students is still negligible.*

Here's why the computer hasn't penetrated the educational system:

- *System reliability.* Without doubt, the one thing that will turn even the receptive teacher away from educational technology is system unreliability. A teacher who, on three consecutive days, brings a film and projector into his classroom, only to have the lamp burn out, a fuse blow, or the projector chew up his film, is unlikely ever to bring a film into the classroom again. The same is true with the computer, whether the fault belongs to the telephone company, the computer manufacturer, the system operator, or some student who crashes the system.
- *Teacher preparation.* The teacher can master film projectors, cassette players, overhead projectors, video tape players, and most other technological devices in minutes, or, at the most, hours; but the computer requires months for mastery – at least in the perception of the teacher. (In reality, mastery of the computer is not required in many cases, as demonstrated by the Huntington Computer Project and the TIES Project.)
- *Cost.* Until recently, the purchase or annual rental of a single port into a computer cost $5,000-10,000. Even the annual rental of a terminal to access a computer which otherwise is free costs $500-1,000. Even in the best of times, such expenditures severely strain school budgets. (Typically, a high-school science department will receive a total annual budget for all purposes of $100-500).

---

* The author's children attend a high school with a 16-terminal minicomputer available. There are only a few school systems in the U.S. with more terminals available, yet, with an enrollment of 2,800 students, 70 terminals would be required to provide only one hour on the computer per student per week.

• *Availability of courseware.* In the typical classroom, teachers cannot take class time or their own limited preparation time to develop computer programs. They have neither the time nor the training, so the availability of high-quality courseware is crucial. Every group and individual I visited emphasized that large amounts of high-quality courseware must be made available. Every computer manufacturer who has serious designs on the educational market is searching actively for courseware authors.
• *Size.* A dozen years ago I used a computer interactively in my classes. Timesharing was required then. For each class I had to move the teletypewriter (weighing over 100 pounds) from the fourth floor to the eighth floor and back. Even stand-alone minicomputers didn't solve the problem. Although they are much smaller and lighter than conventional computers, they still are too bulky and too heavy to move easily.
• *Language.* There's a great deal of confusion about which computer language should be used in an educational environment. A few candidates are: ALGOL, APL, BASIC, COBOL, LOGO, PASCAL, and PILOT. Each language has good and bad points, and each has champions and detractors (both usually are very vocal). The controversy will continue to rage, but Thomas Dwyer advises educators, "Debate about these issues will continue of course, and that's a healthy thing. And someday there may be readily-available super languages. In the meantime, 'enjoy'."

Present status of microcomputers

MITS, which developed the Altair 8800, pioneered a new market into which many companies (mostly small) rushed. Among the "first generation" of microcomputers that survived the initial shake-out, in addition to the Altair 8800, were the Altair 680, the IMSAI 8080, and the Southwest Technical Products 6800. All of these computers are made well and provide adequate computer support for the typical classroom (pre-college or university). All perform at roughly comparable levels, and all cost about $2,500-3,000 for a computer with 16K bytes of random access memory (RAM), a keyboard, a video monitor and interface, and an audio cassette recorder (for mass memory) and its interface.

Let's pause at this point, to salute one of the pioneers in bringing microcomputers into schools – Dr. Peter Grimes of the San Jose Unified School District. Under his direction about a dozen microcomputers have been installed in San Jose schools. These microcomputers have enhanced the district's computing capacity at a cost significantly lower than would have been required to expand the existing mini timesharing system.

Before we describe the characteristics of the newly-available "second generation" microcomputers, let's see how far computing has come since 1945. John Linvill and C. Lester Hogan have compared the ENIAC computer with a Fairchild F8 microprocessor to which has been added 2K bytes of read-only memory (ROM), 8K bytes of RAM, and a teletype interface. This table shows the results:

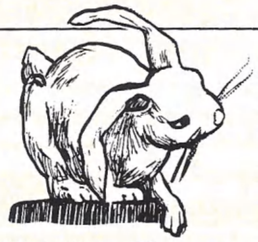
COMPUTERS IMPROVE

	ENIAC	F8	Ratio
Size, cubic feet	3,000	.011	300,000
Weight, pounds	60,000	1	60,000
Clock rate, cycles/sec	100,000	2,000,000	20
Add time, micro-seconds	200 for 12 digits	150 for 8 digits	about equal
Mean time to failure	hours	years	10,000
RAM memory, kilo-bytes	1	8	8

Since 1975, the reliability and capabilities of microcomputers have gradually but persistently improved, and a wide variety of peripheral devices have been developed. In 1977 Commodore Business Machines unveiled the PET 2001 microcomputer. The announcement was dramatic for two reasons:
• The PET was designed with the consumer and the educator in mind, rather than the hobbyist.
• The PET was the first *complete* microcomputer (lots of memory, video monitor, keyboard, and audio cassette recorder, all with interfaces, plus the usual processor and power supply) offered for under $600.

The PET has these consumer-oriented features:
• *Instant-on operation.* There is nothing to load, no switches to set. When the power switch is flipped to ON, the system is ready to be used.
• There is no elaborate, frightening (even though useful to the hobbyist) front panel with its arrays of LEDS and switches.
• *The system is completely assembled.* There are no kits, and essentially no decisions to make (How many serial and parallel I/O ports should I buy? Which of all the available peripherals should I purchase?)
• *Easy maintenance.* The PET has a diagnostic routine loaded in ROM with which (it is claimed) any TV repairman can identify a defective board. The defective board will be sent to a board repair center for repair or replacement. (This capability still must be measured in practice, since no machines have been delivered up to the time of this writing.)
• Almost all the electronic components are mounted on a single printed-circuit board (there are only three total), to minimize the number of interconnections and offer the

Since 1975, the reliability and capabilities of microcomputers have gradually but persistently improved, and a wide variety of peripheral devices have been developed.



possibility of significantly improved reliability over the "first generation" microcomputers.

The PET has a 6502 microprocessor, a 14K ROM (including 8K BASIC, a 4K operating system, a 1K diagnostic routine, and a 1K machine-language monitor), a 4K RAM (an additional 4K costs $200), a 73-key keyboard that prints 64 ASCII characters and 64 graphic characters, a built in

PET Advantages

1. Excellent graphics.
2. Excellent BASIC
   a. String functions.
   b. Floating point and integer arithmetic. Ten-digit accuracy.
   c. File management system, dual cassette, named files.
   d. Software access to external devices.
3. Full cursor control for editing; software control of cursor.
4. Ports to access external devices
   a. Eight-bit parallel port.
   b. IEEE 488 interface.
   c. Port for second cassette recorder.
5. Keyboard advantages
   a. Separate number pad.
   b. All alphanumeric characters plus all mathematical symbols available without shift key.
6. Hardware diagnostic routine in ROM.
7. 8K byte BASIC plus 4K byte operating system.
8. Program execution substantially faster than TRS-80.
9. Data transfer rate for cassette recorder, 1000 baud.
10. Distribution will be through selected computer stores, as well as through national department stores (Sears, Penney's, or Macy's are possibilities).
11. The monitor, recorder, and computer form a single integrated unit which eliminates all cables. The user need only plug the computer into a wall outlet to be on the air.

PET Disadvantages

1. Keyboard too small (key-to-key distance only two thirds that of normal keyboard) for touch typists. Essentially a calculator-style keyboard.
2. Video monitor nine inches rather than twelve.
3. Memory expansions relatively expensive.
4. After first 4K byte memory extension, additional memory expansion requires external case.
5. 40-character line on monitor.
6. Maximum memory, 32 K bytes.
7. PET more bulky than TRS-80.
8. No UL approval.

1,000-baud audio cassette recorder, an IEEE 488 instrument port, an 8-bit parallel port, a port for second audio cassette recorder, and BASIC with 10-digit accuracy, full cursor-controlled editing, software control of the cursor, file management on dual cassettes, and floating point plus integer arithmetic. The whole PET is only 18¼" long, 16½" wide, and 14" high (including the monitor) and weighs 44 pounds.

Commodore was the first company to announce a personal consumer computer, but it hardly will be the last. In July 1977, Radio Shack (via Tandy Computers) announced a competitor of the PET. In addition, there are persistent rumors that Atari (the video-game pioneer), Texas Instruments, Coleco, and several chip-set manufacturers will join in the competition. The Radio Shack TRS-80 computer has a Z-80 microprocessor, 4K ROM containing level 1 BASIC, 4K RAM (4K addition costs $99.95, and a 12K addition costs $289.95), a standard 53-key keyboard, and a 300-baud interface for external audio cassette recorder.

The BASIC is very minimal. It has few built-in functions. It has no string commands, commands to access external ports, dimension statements, or user-defined functions. It does have simple graphic characters (128 horizontal x 48 vertical). The price is $399.95 for the computer only, or $599.95 for the computer plus 12-inch black-and-white monitor and audio cassette recorder.

There has been an interesting casualty in the new genre of computer. Neither the PET nor the TRS-80 nor the other computers I'll describe uses the S-100 bus, which the Altair 8800 uses for interconnecting computing components, and which until now was the standard bus for the microcomputer industry. Many peripherals have been developed for that bus. At first glance, the rejection of the S-100 bus appears to be very unfortunate, because you can't use those peripherals with the new computers. Although this is true, it may only be a short-term problem. If the PET expectation of selling 10,000 units per month is even roughly realistic, within six months there will be twice as many PETs as all other microcomputers combined and the PET bus will become the de facto standard. Because most companies marketing microcomputer peripherals are small, and consequently very flexible, they undoubtedly will respond quickly to the PET invasion, if it should occur.

Graphics

The Intecolor 8051 and the Apple II offer color graphics at a reasonable price. (See chart.) The first color graphic device available for microcomputers was the Cromemco TV Dazzler, a single board that plugs into the S-100 bus, has 128 x 128 picture elements and allows eight colors in either of two intensities.

Another entry into the graphics arena was the Merlin terminal, which you can use with most microcomputers: it has a black-and-white video monitor, displays 20 lines of 40 characters, and handles graphics; the graphics capability is addressable through BASIC and has two plotting-density options: 160 x 100 grid (total cost $349) and 320 x 200 grid (total cost $403).

During the past decade, Dr. William Huggins of Johns Hopkins University has been telling the world (and demonstrating to it) how important graphic information is in learning. Thanks to him and others (notably Drs. Zajac, Knowlton, Sinden, and Noll of Bell Telephone Laboratories, and Dr. Bork at University of California at Irvine), it is now clear *how* to use graphics, in at least some applications. But only within the last two years have educators been able to *afford* graphic devices. Now the rest of us can follow the trail that has been broken by these pioneers.

The question of intelligent use of the color offered by Dazzler, Apple II, and Intecolor remains. Many people feel intuitively that the addition of color will substantially increase the educational impact of computer graphics; but the large cost of such systems in the past has prevented serious investigation of color's effectiveness.

Fancy peripherals

Besides graphics systems, the new electronics cottage (perhaps *garage* is a better word) industry has produced other peripheral devices that have educational potential. The most immediately-applicable new peripheral in the learning environment is Cromemco's seven-channel D/A and A/D converter, which costs under $300. Many educational games (such as Lunar Lander) and simulations (such as Charge – the Millikan Oil Drop simulation) will become more realistic for the student. Young children can be trained to develop eye-hand coordination. Through electromechanical sensors and effectors, students will be able to control directly processes for which they do not have the training or the dexterity. In addition, the digital computer can be converted essentially to a very flexible analog computer (without the drift or scaling problems of real analog computers.)

In the Fall of 1976, two companies announced single-board speech generators which cost under $400, are plug-compatible with most microcomputers, and produce human-like speech by stringing together phonemes under program control. My grandsons are fascinated by the Telesensory Systems, Inc. Talking Calculator. An acquaintance of mine is exploring this calculator's impact on mentally-retarded children; the preliminary results are encouraging.

In 1977 speech-recognition boards were announced. You can teach Heuristics Speech Laboratory's board to recognize any number of spoken words (depending on available memory); the board costs $299 assembled. The mind boggles at the thought of a computer that can *interact* with students *verbally.*

One of the most impressive projects at the Student Fair held at the 1976 National Computer Conference was a system consisting of a Wang Computer and a $2,000 Votrax speech synthesizer, which presented arithmetic drill verbally. It was the hit of the Fair and made that evening's TV news broadcast.

You can enter visual information into a computer by using Cromemco's Cyclops camera. The Cyclops consists of a lens, an electro-optic sensor matrix in a 32 x 32 array, and an interface board to convert the sensor signals into a form that can be dumped into the computer's memory. (My students are exploring the possibility of using this system to help blind people read.)

Several companies have produced peripheral boards for generating music. The simplest devices just string notes together into melodies; others permit the programmer to control attack, decay, and timbre of the sound in complex ways.

Almost every month, *Byte, Personal Computing, People's Computer Company,* or *Creative Computing* announces a new peripheral device to do something no one ever considered before. Some of these devices are aimed at the computer hobbyist and some at the computer professional. Many of these are of potential interest to the educator, but most need development (at the level of the educational system rather than at the peripheral level).

Is timesharing dead?

Another development of interest in educational computing is the emergence of timeshared microcomputer systems. At least two computer manufacturers (Altair and Cromemco) have announced timeshared microcomputers. It is impressive that people are able to accomplish this; but I wonder to what extent it represents doing something just because it can be done. One of the significant benefits microcomputers from the Altair to the PET have brought to educational computing is the freedom from the cost and unreliability problems of communicating with the timeshared computer, and from the tyranny of the young computer freak who can crack any security system and get into other peoples' files for the thrill of mucking around where he doesn't belong.

Radio Shack TRS-80 Advantages

1. Professional-quality keyboard.
2. 64-character line on monitor.
3. 12-inch video monitor.
4. Memory expansion capability to 12 K ROM and 16K RAM *internally.*
5. Inexpensive memory expansion.
6. Approved by Underwriters Laboratories.
7. System may be purchased without monitor or recorder (if purchaser already has these units).
8. Computer is about one-quarter volume of PET excluding monitor.
9. Maximum memory, 65K bytes.
10. Distribution will be through 146 Radio Shack stores nationwide.

Radio Shack TRS-80 Disadvantages

1. Level 1 BASIC is relatively weak.
2. Less versatile graphics.
3. Lower rate data transfer on cassette recorder.
4. Significantly slower program execution than the PET.
5. Monitor and recorder must be connected to computer by external cables before computer becomes operational.

In my judgment, timesharing is essentially dead in educational computing. My position may seem extreme; but it's correct, for these reasons:

• *Sharing cost of computer hardware.* With computer-system costs below $600, costs no longer need to be shared, especially when you consider that the annual cost of rental of a telephone line plus a teletypewriter exceeds $600.

• *Access to large data bases.* Until recently, only timesharing and batch-processing machines could handle large data files. The dual-cassette file management system of the PET, inexpensive mini-floppy disc systems costing less than $1,000, and video-disc systems make such access relatively easy to afford in the stand-alone system.

• *Interpersonal communication.* Accessing the files of others, leaving messages, and reading messages from others in your "community" is useful and intellectually powerful. In addition, interpersonal communication lets you implement complex multi-person simulations. That capability is easy with a timeshared system, but also possible with stand-alone microcomputers equipped with modems. (The Pennywhistle modem is an inexpensive device here). An experiment in such interpersonal communication, called PCNET (Personal Computer Network), was starting in the fall of 1977 in Palo Alto. Here a person wishing to communicate with another tells the computer to store the message in the second person's computer memory.

Intercolor

a. $3,995
b. 8080 microprocessor
c. ROM, 11K bytes, expandable to 32K bytes
d. RAM, 16K bytes, expandable to 32K bytes
e. 19 inch color video monitor
f. 80 characters/line x 48 lines, upper and lower case
g. Graphics
  I. 8 foreground and 8 background colors
  II. Plotting on 160H x 192V grid
  III. Software available for vector, bar plot, and point plotting
h. Powerful editing
i. Optional light pen ($675 additional)

Apple II

a. $1,800 (for 16K byte RAM)
b. 6502 microprocessor
c. ROM, 8K bytes, expandable to 12 K bytes
d. RAM, 16K bytes, expandable to 48 K bytes
e. No monitor or cassette recorder supplied
f. 40 characters/line x 24 lines, upper case only
g. Graphics
  I. Plotting on 40H x 48V grid, 15 colors.
  II. High-resolution plotting: 280H x 192V grid, 4 colors
h. 1500-baud cassette interface
i. Game interface with two joysticks
j. 6 K BASIC plus 2 K monitor in ROM

An open letter to the Chicago Show Thief:

Your judgment is excellent, your morals are deplorable,
you overlooked one little thing... and you've got a problem

We think we know just how it went --

You paid your way into the Personal Computing Show, there at the Chicago Holiday Inn-O'Hare in late October. You moved unnoticed among the wall-to-wall crowd, browsing through the fascinating exhibits -- dozens of microcomputers and their peripherals, all fascinating, valuable stuff. You came to the separate room where our "professor" Gene Murrow was conducting his classes in Basic programming. And you watched with awe and envy as Gene put a Compal-80 system through its paces, showing off its versatility -- as a word processor... text editor... as an accounting tool, doing its real estate investment analysis work... as a system for the highly-specialized task of planning the adaptation of a theatrical motion picture to the special editing needs of TV broadcast... You cased the joint and crept away, planning your shabby little plans.

And sometime under cover of darkness before dawn on Saturday morning, you tiptoed back into the Computer Power & Light classroom, passing up the dozens of units closer to the door and easier to haul away. You put your sticky hands on our demonstrator setup: the computer, the mini-floppy disk drive, the keyboard, even the manual. And you sneaked off into the night with your guilty burden.

In one respect, you're not so dumb. You made a point for us: when you have the cream of America's microcomputer systems from which to make a selection, the choice is the COMPAL-80 system. Right you are. Wrong to steal it, right to choose it.

Just one thing: you failed to steal the disk that will let you use it. There's no way to operate that system without Compal's own extended version of disk Basic. And if you ask us for that, we'll call the cops.

$100 REWARD
FOR INFORMATION LEADING
TO THE RETURN OF THE LOOT

Compal will pay a handsome ONE HUNDRED BUCKS for the tip that lets us recapture our $3800-worth of stolen goods -- useless, as we say, without that special Basic (and if you think we won't be on the alert for anybody trying to buy the Basic by itself, think again).

The items:

* Compal-80 computer (S-100 bus, with 32K memory, CPU, video board I/O interfaces) Serial #800877004
* North Star Disk Drive
* Compal Keyboard (upper and lower case, 10-key pad)

Call or write in confidence -- to our Studio City store.

COMPUTER™ POWER & LIGHT INC.

12321 Ventura Blvd., Studio City, CA 91604 / (213) 760-0405

7878 Clairemont Mesa Blvd., San Diego, CA 92111 / (714) 565-2041

CIRCLE 15

Are inhibitions gone?

Let's review the factors that have inhibited the entry of the computer into the educational environment, and determine the extent to which microcomputers can dispell them. The factors are:

- *System Reliability.* The microcomputer promises to significantly increase the reliability of computer systems. Because of large-scale integration and the trend toward single board computers, interconnections are being made permanently and internally, resulting in few failures in connections. Because timesharing is no longer necessary, we can eliminate the unreliability caused by telephone lines, modems, timesharing hardware, and computer freaks.
- *Teacher Preparation.* Clearly, no development we've mentioned has any effect on this problem area, except perhaps by providing motivation to the teachers.
- *Cost.* The PET and the Radio Shack TRS-80 computers have made dramatic inroads on eliminating this problem.
- *Availability of Courseware.* Although we haven't unveiled any development that addresses this inhibition directly, the promise of selling 100,000 computers per year in the immediate future has attracted the attention of many potential courseware authors. This attention should help alleviate the courseware problem.
- *Size.* We still don't have quite the briefcase-size computer, but the PET and the TRS-80 are very portable. This portability is important in education, because you can now bring the computer into any classroom having a wall outlet. Students can take computers home for a night or a weekend. Teachers can take computers to their offices or homes to develop their own expertise in privacy. One can imagine local libraries circulating PETs to their patrons. (My own local library has two timeshared computer terminals for use by local residents.)
- *Language.* None of the developments we've mentioned has impacted language development.

Future

"We see through a glass, darkly." It is difficult also to predict the classroom's character a decade hence, or the role of the computer in it. (Some visionaries predict the classroom will *disappear* within a decade — or two at the most — and that all learning will take place in the individual home, and the computer will play a central role. Although that outcome is possible, it isn't likely.)

The microcomputer era upon which we are embarking already has been characterized as the era of the personal computer. It certainly will be, but it promises to be much more. Educational and social historians someday may look upon January 1975 as the start of when the computer enabled humans to expand their intellectual powers beyond our present comprehension.

Theodor Nelson predicts that by 1980 computers will enter 10 million American homes.

Will there be another dramatic breakthrough next month? in six months? two years? It is impossible to guess, but whenever it happens educational computing will make another large stride forward.

What Should You Charge
For Computer Services

BY ELIZABETH M. HUGHES

As an entrepreneur, you'll not only be hard-pressed to learn about the business needs of your clients, but often at a loss to know what your own business should be doing and charging. This discussion provides no magic formula, but offers a useful approach to mysteries of pricing.

Once your system is up and running, the catalogs and new product announcements take on a new poignancy. If only you could afford that neat peripheral – or that extra 8K of memory! This reflection turns most hobbyists' minds, sooner or later, to the idea of using the computer to raise money for those improvements. You look around the neighborhood and see lots of people who have a crying need for computer services (whether they know it or not); you find magazines filled with programs for saleable services; and you say, "What am I waiting for?"

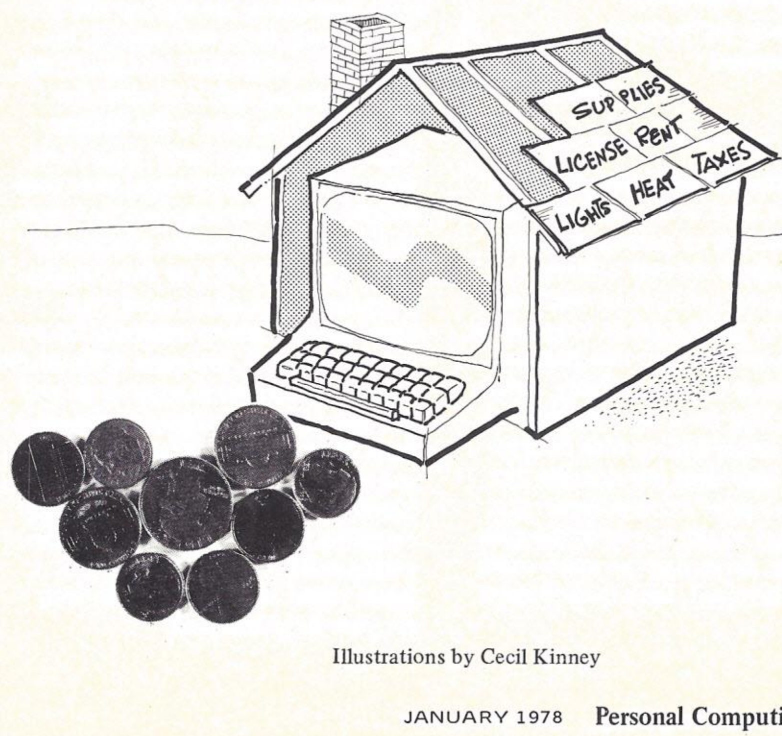
If you're going to succeed in the business, though, several things are necessary: a growing library of saleable services that you *know* you can perform satisfactorily (committing yourself to a project you've never tried before is tempting but can lead to disaster); a way to attract customers to your product and a fair scale for pricing your services. A few test runs using sample data will let you offer with confidence the services published in the magazines, providing you with an initial portfolio of programs. As your familiarity with and confidence in other programs grow, they can be added to your catalog of services. Once you have something to sell, selling clients is usually a question of showing them how they'll benefit from it. Frequently, a good service will practically sell itself if you can show your prospective customer how it reduces work.

Most important, however, is the service's pricing. If you price the service too low, you'll lose money; if too high, you'll lose – or never acquire – customers. The basis of any successful business is a satisfied clientele, and in these days of appalling inflation and tight money, an essential to maintaining your clientele is to price your services fairly. Most computer entrepreneurs, however, hardly know where to start when it comes to figuring prices, since only a few of us have useful business experience. The object of this article, then, is to give you some facts and figures that can be price guidelines and a few rules of thumb for meeting the pricing needs of your particular community.

What's overhead?

To run your business you must deal with certain unavoidable expenses called *overhead;* they include the costs of your place of business (heat, rent, lights, etc.) and any business licensing fees. Ask your local government about its business requirements, since in some towns you may conduct your business only in certain zones or must pay a licensing fee annually. Sometimes there are other requirements. Many states and municipalities require that you collect a sales tax on products and services.

Getting an answer from government seems to entail many phone calls. A, whom you called first, tells you to call B who tells you to call C who tells you to call D who, after failing to convince you to call A again, gives you the answer. Go into the process armed with pencil and paper for notes and as much



Illustrations by Cecil Kinney

patience as you can muster. The federal government seems to answer requests and inquiries more quickly and with less runaround than local governments, but if you always expect the worst, you'll sometimes be pleasantly surprised. After the phone calls, you might might have to drive around – fill out forms and pay license fees – but the preliminary call will assure that you make no unnecessary trips.

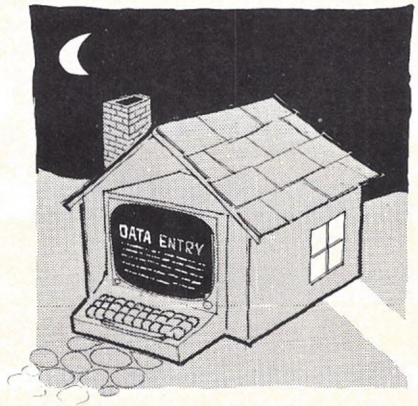
Once you've satisfied government prerequisites, figure your overhead. Traditionally, overhead includes every expense which, while necessary to your business, does not represent an item for resale; but most computer entrepreneurs are in computing primarily for fun and only secondarily to make money. For this reason, you'll probably prefer to leave out of your calculations anything which isn't just a business expense. Thus, your home (which is not just used for your office), your computer (which is used for fun as well as business), and your car (which is not used solely for delivering printouts) would probably be ignored even though, if you become a small business in the eyes of the Internal Revenue Service, part of these expenses would be legitimately deductible as business expense. Your list of business-only overhead expenses will probably look more like this:

BUSINESS OVERHEAD

Business license fee	$25.00
Sales-tax collection license fee	10.00
Program purchased solely for business purposes	5.00
	$40.00

This is cost you would have avoided if you hadn't tried to be a business. You recoup that loss by pricing your services enough above cost to cover it.

Besides recovering your overhead, you want to make a profit to spend on your system. Don't get into business if it will result in losing money, unless you're looking for a tax loss. Most of us would rather end up with cash to spend on upgrading our systems. But in trying to price a computer service, it's too easy to think only of the computer time, forgetting the many other expenses involved. Let's look at these expenses . . .



Entering data

Most of the time spent performing a computer service is usually devoted to entering the data. Since few computer entrepreneurs are experienced data-entry operators – and fewer still can afford to pay professionals to do the job – data entry can be long and painstaking.

Salaries for professional keypunch operators vary; without professional keypunch or typing experience you would most likely be paid minimum wage. But as an entrepreneur, you can reasonably ask for $3 an hour, because you could have used that time to improve your own business (by programming, advertising, or whatever), and because you should be compensated for the pain and nuisance of entering all that data. After spending ten hours puzzling out someone's abysmally handwritten data, you'll see why you deserve at least that much.

Computer time

If your programs are written and entered already (you may want to include that cost in your overhead), the only remaining task is to enter the data. Computer time is the hardest single item to price fairly, since it reflects local electricity rates, your computer's power consumption, and wear-and-tear on the machine. Since local power rates vary and no two computer systems, even of the same type, have identical power consumption or breakdown rates, pricing is difficult. So call up a local time-sharing outfit and find out what *it* charges for computer time. Then undercut them. On these and similar grounds, I use 5¢/minute ($3/hour) as my basis for calculating computer time both in my own computing and in this article; but that figure might be too low or too high for your neighborhood.

Your supplies

Once the data's been processed, you must output it in a form the customer can use. That introduces the expense for supplies – usually second only to data entry as an item on the bill. Fortunately, supplies fall straightforwardly into two categories – reusable and non-reusable – and can be priced easily.

A typical non-reusable supply is printout paper. You can buy a 500-foot roll of Teletype paper for about $2.50. Double that, to allow for the initial expense of the paper and a small profit margin, and it becomes reasonable to charge 1¢/foot of paper used on the project, calculated on the basis of the *total* paper the project requires, not merely the part given to the customer as a printout. You can price other non-reusable computer supplies (such as coding sheets) analogously.

Door-to-Door Sales, Inc.
(Order/Commission Records)

Set-up Charges	
Programming	
5 hours @ $5	$25.00
Data entry	
3 hours @ $3	9.00
Computer time*	
45 min. @ 5¢	2.25
*(actual run time plus dump and load time, in all cases)	
Data & program storage	
1 cassette	5.00
Output	
25 ft. @ 1¢	.25
Total set-up charge	$41.50
Update Charge	
Data entry	
2 hrs. @ $3	$6.00
Computer time	
15 min. @ 5¢	.75
Output	
10 ft. @ 1¢	.10
Total update charge	$6.85

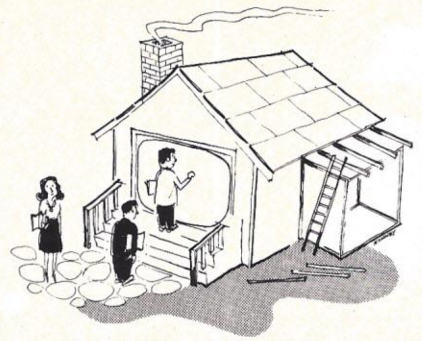
Reusable supplies, such as tape cassettes or floppy disks, pose a different problem, since *you*, not your customer, will benefit most from their purchase. After the conclusion of a project it is often beneficial to keep your customer's records on file (3 months to 5 years, with a common waiting period of one year); but eventually you can erase the tape or disk and re-

use it for another project or customer. For the first cassette or floppy a project requires, it's reasonable to charge about twice what it cost you: charge about $5.00 for a cassette, $8.00 for a mini-floppy, and $10.00 for a standard floppy. Those figures apply if the whole reusable device is for a single customer's project. If you're storing more than one customer's material on a single tape or floppy, split this charge between them in a way that reflects how much it is used for each of their projects.

Downtown Bowling League	
(Scores and Other Records)	
Set-up Charges	
Program use	
In library	$1.00
Data entry	
6 hours @ $3	18.00
Computer time	
30 min. @ 5¢	1.50
Data & program storage	
2 cassettes	8.00
Output	
50 ft. @ 1¢	.50
Total set-up charge	$29.00
Update Charge	
Data entry	
3 hrs. @ $3	$9.00
Computer time	
30 min. @ 5¢	1.50
Output	
23 ft. @ 1¢	.23
Total update charge	$10.73

If a job requires more than one cassette or floppy, give the large customer a price break (unless he insists on the possession of the relevant tapes at the end of the project). I drop my charge to about $3 for the second cassette, $5 for the second mini-floppy, $6 for the second standard floppy, and sometimes even lower when three or more cassettes or floppies are used. The big customer deserves a price benefit because the project's size assures you an income over a longer period of time and, if satisfied, the customer may recommend you to other large users.

Other external storage media and other types of reusable supplies can be similarly priced; but if you must



make a major equipment investment in order to do a job, the customer should not be expected to bear the brunt of its cost. Equipment investments in the long run will benefit you far more than the customer. You can legitimately add 10-25% of the equipment expense to the customer's bill, but any such arrangement must be worked out with your client in advance. Don't add a whopping equipment charge to a bill without giving the customer fair warning and a chance to back out. You can charge as much as 25% if the client will be almost the sole user of the item and you expect little personal benefit from adding it to your system; but charge 10% (or less!) if you expect much personal use from the item or if several customers will benefit from its purchase. This can create a situation in which you must turn down a project because you can't afford the expensive plotter (or whatever) it requires; but a customer can understand and appreciate your inability more easily than an unexpected bill. Discuss all such problems with your customers before giving flat answers, since sometimes a customer will be willing to underwrite the equipment purchase necessary for the job in exchange for a certain quantity of computer services. In effect, the customer buys the item from the supplier for cash and resells it to you for cash and/or services.

Charge for programming

Sooner or later you'll have to write your own programs or modify a program already in your library. Since that requires your special skills – not merely your special equipment, the computer – feel free to charge $5.00/hour if you're a competent non-professional. Professionals should charge more; beginners, less. The customer who requested the new or improved service pays the programming cost. Thereafter, the program or improvement becomes part of your library. Charge nothing or a very nominal fee (e.g., $1.00) to use programs in your library.

Itemize your prices

Although you may rarely want to itemize the charges when you bill a customer, itemizing for your own benefit is the only way to be sure your prices reflect the services rendered in each given instance. To reach the final price, see the two sample computations.

These figures give some indication of how projects may differ. In some, the bulk of the work will be done in the set-up and the updates will be easy; in others, the set-up will be trivial and the update will be troublesome.

Estimate the cost

Few customers will commit themselves to a project when they have no idea of what it will cost. But only experience can give you a basis for estimates. Get some of the experience you need when you're familiarizing yourself with the service. Instead of just entering a program and immediately trying to sell it, run it awhile using data from imaginary customers. Find out how long it takes you to enter different amounts of data, how much machine time and output it involves, and anything else you can. Then, when a prospective client asks how much your service will cost, you can give a reasonable estimate of the price.

Final Thoughts

Since your system and your skills are a little different from everyone else's, you must make your own decisions about what to charge, although the prices I've discussed here can serve as guidelines.

The key to pricing in all businesses is summed up in that wonderful phrase "all the traffic will bear." If a given neighborhood won't pay a $30-40 set-up charge and a $10 update charge, charge less. You can charge more, if they'll pay. You're selling the advantages and convenience of dealing with an individual instead of an institution; but you must also offer a lower price. If a suspicious customer can call any local computer business and either get the brush-off because the job is too small or the estimate looks outrageous next to yours, that customer will come to you – and tell his friends. Then you can afford the next addition you want for your system. PC

ILLUSTRATING BASIC

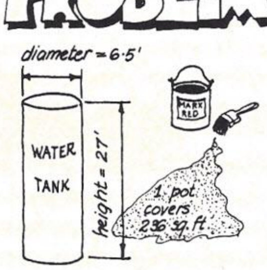
(A SIMPLE PROGRAMMING LANGUAGE)

DONALD ALCOCK

"*BUG*" IS COMPUTER JARGON FOR A MISTAKE. THE BOOK SHOWS THIS LITTLE BUG AGAINST ILLUSTRATIONS OF MISTAKES IN PROGRAMS.

PROBLEM

HOW MANY POTS OF PAINT DO YOU NEED TO PAINT THE ROOF AND WALL OF THIS WATER TANK?



WE COULD GO STRAIGHT AT IT LIKE THIS:

roof area, T = π×6·5²÷4 = 33·2
wall area, S = π×6·5×27 = 551
total area, A = T+S = 584·2
number of pots, G = A÷236 = 2·48
rounding up, R = 3
∴ you need 3 pots of paint

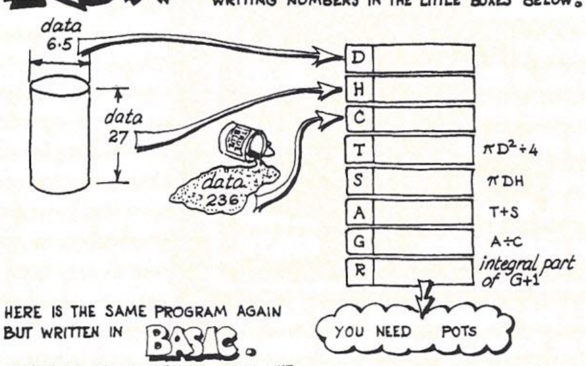
OR WE COULD WRITE A ***PROGRAM*** (IN ENGLISH) TO SOLVE THE PROBLEM.

*1. **REMARK**: A PROGRAM IN ENGLISH*
2. THE FOLLOWING NUMBERS ARE THE **DATA** 6·5, 27, 236
3. **READ** THE DATA, CALLING THEM **D, H & C** RESPECTIVELY
(think of this as putting the data into little boxes labelled D, H & C respectively – see opposite page)
4. WORK OUT 3·14×D²÷4 AND LET THE RESULT BE CALLED **T**
(i.e. put the result in a little box labelled T)
5. WORK OUT 3·14×D×H AND LET THE RESULT BE CALLED **S**
6. ADD **T** TO **S** AND LET THE RESULT BE CALLED **A**
7. WORK OUT A÷C AND LET THE RESULT BE CALLED **G**
8. ROUND **G** TO THE NEXT WHOLE NUMBER AND LET THE RESULT BE CALLED **R**
(i.e. add 1 to G and take the integral part of the result)
9. ***PRINT*** "YOU NEED"; **R** ; "POTS"
(i.e. print whatever whole number R turns out to be)
*10. THE **END***

THIS HAS THE ADVANTAGE OF BEING GOOD FOR ANY SIZE OF TANK AND PAINT POT & YOU NEED ONLY REPLACE THE DATA ON LINE 2.

NOW

TRY *OBEYING* THE ENGLISH PROGRAM OPPOSITE & FEEL WHAT IT WOULD BE LIKE TO *BE* A COMPUTER & *DEFILE* THIS PAGE BY WRITING NUMBERS IN THE LITTLE BOXES BELOW.



HERE IS THE SAME PROGRAM AGAIN BUT WRITTEN IN BASIC.

COMPARE IT CAREFULLY WITH THE *ENGLISH* VERSION OPPOSITE.

```
1   REM   A PROGRAM IN BASIC
2   DATA  6.5, 27, 236
3   READ   D , H , C
4   LET    T = 3.14*D↑2/4
5   LET    S = 3.14*D*H
6   LET    A = T+S
7   LET    G = A/C
8   LET    R = INT (G+1)
9   PRINT  "YOU NEED"; R ; "POTS"
10  END
```

notice
* meaning multiply
↑ meaning raise to a power
/ meaning divide

AND THIS, WHEN OBEYED, WOULD PRODUCE:

```
YOU NEED 3 POTS
```

FIRST

PREPARE YOUR PROGRAM BY TYPING INSTRUCTIONS AT THE KEYBOARD ⇒ THE COMPUTER SIMPLY *STORES* THE PROGRAM AT THIS STAGE :

IT DOESN'T OBEY ANY INSTRUCTIONS

```
1 REM  A PROGRAM IN BASIC
2 DATA 6.5, 27, 236
3 READ  D, H, C
4 LET T=3.14*D↑2/4
5 LET S=3.14*D*H
6 LET A=T+S
7 LET G=A/C
8 LET R=INT(G+1)
9 PRINT "YOU NEED";R;"POTS"
10 END
```

THEN

TYPE RUN

WHICH SETS THE COMPUTER TO WORK *OBEYING* THE STORED INSTRUCTIONS ONE AFTER THE OTHER ⇒ IN NUMBERED SEQUENCE ⇒ WHILST YOU RELAX.

EVENTUALLY THE COMPUTER WILL OBEY THE INSTRUCTION END. THAT MAKES IT STOP.


```
RUN
YOU NEED 3 POTS
```

BUT

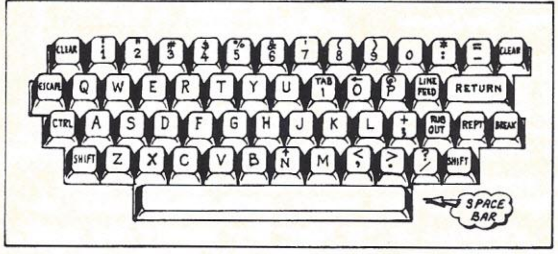
BEFORE YOU CAN TAKE THE FIRST STEP AND START TYPING THE PROGRAM YOU HAVE TO GO THROUGH THE RITUAL OF *SIGNING ON* AND TELLING THE COMPUTER YOU WANT TO USE *BASIC*.

DIFFERENT COMPUTERS (EVEN IDENTICAL COMPUTERS RUN BY DIFFERENT ORGANISATIONS) OFTEN HAVE DIFFERENT WAYS OF DOING THESE THINGS, SO IF YOU WANT TO TRY THE PROGRAM NOW GET SOMEONE WHO "KNOWS THE SYSTEM" TO SIGN ON FOR YOU AND CALL UP *BASIC*.

KEYBOARD

EVERY PROGRAM IN *BASIC* HAS TO BE TYPED ON A KEYBOARD

⇒ *PROBABLY SOMETHING LIKE THIS* ⇒



ALTHOUGH POSITIONS OF LETTERS & DIGITS ARE THE SAME ON MOST KEYBOARDS, KEYS LIKE [RUB OUT] & [BREAK] IN THE PICTURE ABOVE VARY IN NAME, POSITION AND FUNCTION FROM ONE INSTALLATION TO ANOTHER.

NOTICE ALL LETTERS ARE CAPITAL LETTERS. NOTICE ALSO THERE IS A KEY FOR 1 AND A KEY FOR ZERO (BOTH IN THE TOP ROW). NEVER PRESS THE LETTERS I AND O IN THEIR PLACE.

AS ON AN ORDINARY TYPEWRITER, PRESSING [SHIFT] AT THE SAME TIME AS ANOTHER KEY GIVES THE CHARACTER SHOWN ON THE UPPER HALF OF THAT KEY: THUS [#/3] TOGETHER WITH [SHIFT] GIVES # WHEREAS [#/3] ALONE, 3.

THE "BACK ARROW" ← SERVES TO DELETE THE CHARACTER ON ITS LEFT FROM THE COMPUTER'S MEMORY; TWO OF THEM DELETE THE PREVIOUS TWO CHARACTERS, AND SO ON. THUS IF YOU TYPE PRIMP←←NT THEN *BASIC* RECEIVES THE WORD PRINT. (REMEMBER THIS BY EXCLAIMING *OH SHIFT!* WHEN YOU HIT THE WRONG KEY.) SOME *BASICS*, HOWEVER, USE AN UNDERSCORE CHARACTER FOR THIS PURPOSE: PRIMP__NT.

MOST *BASICS* USE A KEY (PERHAPS "RUBOUT") WHICH, WHEN PRESSED, DELETES THE WHOLE OF THE LINE YOU ARE TYPING FROM THE COMPUTER'S MEMORY; ANOTHER (PERHAPS "BREAK") STOPS A PROGRAM RUNNING.

FOR A *NEW LINE* IN *BASIC* PRESS [RETURN] (WHEN TYPING "OFF LINE" PRESS RETURN then LINEFEED)

TYPING

IF YOU INTEND TO USE *BASIC* A LOT, LEARN TOUCH TYPING. TEN FINGERS ARE FASTER AND LESS FRUSTRATING THAN TWO.

THERE IS A LIMIT TO THE LENGTH OF A TYPED LINE ⇒ MOST *BASICS* ALLOW LINES UP TO 72 CHARACTERS LONG. SOME ALLOW LONGER LINES BUT IT IS BEST TO ACCEPT A LIMIT OF 72.

SOME *BASICS* ALLOW GREAT FREEDOM WITH THE SPACE BAR: SOME DISREGARD SPACES EXCEPT THOSE BETWEEN QUOTATION MARKS. THUS IT WOULD BE ALLOWABLE TO TYPE:

```
8FORD=STOP
```

INSTEAD OF:

```
8 FOR D = S TO P
```

BUT IT IS OBVIOUSLY SILLY TO OBSCURE THE MEANING OF THE PROGRAM IN ORDER TO SAVE A FEW TAPS ON THE SPACE BAR.

SOME *BASICS* REFUSE TO ALLOW SPACES WITHIN THE CONTROLLING WORDS OF THE LANGUAGE. THUS THE FOLLOWING WOULD BE WRONG:

```
23 L E T A = B+C
```

SOME *BASICS* DEMAND AT LEAST ONE SPACE BEFORE EACH CONTROLLING WORD, OR AFTER IT, OR BOTH :

```
20DATA  6.5, 27, 236
80 PRINT"YOU NEED";R;"POTS"
```

SOME *BASICS* REFUSE TO ACCEPT SPACES WITHIN LINE NUMBERS BUT DO NOT OBJECT TO THEM INSIDE OTHER NUMBERS:

```
1 000 LET A=1 000 . 0
1000 LET A=1000.0
```

SOME BASICS OBJECT TO THESE TOO

SOME *BASICS* DO NOT ALLOW SPACES IN FRONT OF LINE NUMBERS:

```
 95 LET A=B
100 LET C= D*F +G
```

SPACES OPTIONAL HERE

GENERALLY WHEN *ONE* SPACE IS ALLOWED (OR DEMANDED) THEN *SEVERAL* ARE ALLOWED. AND GENERALLY A SPACE IS OPTIONAL ON EITHER SIDE OF THESE ⇒ (, ; * + / − = ↑ > <) BUT NOT IN 1.5E2 (SEE PAGE 9) NOR BETWEEN > AND = (SEE PAGE 41).

A PROGRAM WHICH ACCEPTS ALL THESE RESTRICTIONS SHOULD BE ACCEPTABLE TO ANY VERSION OF *BASIC*.

LINE NUMBERS

LEAVE GAPS IN THEIR SEQUENCE THUS:

```
10 REM   A PROGRAM IN BASIC
20 DATA  6.5, 27, 286←←36
30 READ  D, H, C
40 LET   T = 3.14 * D↑2/4
50 LET   S = 3.14 * D * H
60 LET   A = T + S
70 LET   G = A/C
80 PRINT "YOU NEED"; R ; "POTS"
90 END
```

USE 10s OR 5s

THIS IS 236, SEE PAGE 5

THERE IS A *MISTAKE* IN THIS PROGRAM: THE LAST LET WAS FORGOTTEN. INSERTING IT IS SIMPLE ; JUST TYPE:

```
75 LET R=INT(G+1)
```

AND THE COMPUTER PUTS LINE 75 BETWEEN LINE 70 & LINE 80. IT MAKES NO DIFFERENCE IN WHAT ORDER YOU *TYPE* THE LINES: THE COMPUTER SORTS THEM INTO ASCENDING ORDER OF LINE NUMBER.

IF YOU TYPE SEVERAL LINES WITH THE *SAME* LINE NUMBER THE COMPUTER OBLITERATES EACH PREVIOUS VERSION THUS ACCEPTING THE LINE TYPED LAST. IF THE LINE TYPED LAST IS *JUST* A LINE NUMBER WITH NOTHING AFTER IT THEN THE *WHOLE LINE VANISHES* FROM THE COMPUTER'S MEMORY ⇒ INCLUDING THE LINE NUMBER. THAT IS HOW TO DELETE UNWANTED LINES. THUS:

```
120 LET A=B+C
125 LET E=F
120 LET A=B+G
125
120 LET A=B
```

RESULTS IN THE COMPUTER REMEMBERING ONLY:

```
120 LET A=B
```

THE FIRST LINE NUMBER IN A PROGRAM MUST BE GREATER THAN 0. THERE IS ALWAYS A LIMIT TO THE HIGHEST LINE NUMBER: SOME *BASICS* STOP AT 9999, SO IT IS BEST TO ACCEPT THIS AS THE LIMIT.

THE *LAST* STATEMENT OF EVERY PROGRAM MUST BE: **END**
(NO OTHER STATEMENT BUT THE LAST MAY SAY END).

STATEMENTS

A *BASIC* PROGRAM IS A SEQUENCE OF NUMBERED LINES CALLED *STATEMENTS*.

A STATEMENT MAY SIMPLY *STATE* SOMETHING

```
110 DATA 1, 2, 4
120 END
```

```
30 READ A,B,C
40 LET G = A*B↑2 + C
50 PRINT "ANSWER IS"; G
```

OR IT MAY *INSTRUCT* THE COMPUTER TO *DO* SOMETHING. A COMMON SYNONYM FOR *STATEMENT* IS *INSTRUCTION*; THE STATEMENTS THAT *DO* THINGS ARE *EXECUTABLE* INSTRUCTIONS.

THE COMPUTER FINDS OUT WHAT IS STATED OR WHAT TO DO BY LOOKING AT THE FIRST WORD: DATA, END, READ, LET *etc.*

OR SOMETIMES AT THE FIRST *TWO* WORDS: MAT READ, MAT PRINT *etc.* ((WE MEET MAT ON PAGE 76)).

BUT THERE IS AN IMPORTANT EXCEPTION:

THE WORD **LET** MAY BE *OMITTED* IN MOST VERSIONS OF *BASIC*.

```
40 G = A*B↑2 + C
```

REM

REM STANDS FOR *REMARK*. REM STATEMENTS CAUSE NO ACTION BY THE COMPUTER; YOU INCLUDE THEM TO CLARIFY YOUR PROGRAM.

```
10 REM        *** WATER TANKS ***
20 REM
30 REM   A PROGRAM TO ILLUSTRATE BASIC
40 REM   -*-*-*-*-*-*-*-*-*-*-*-*-*-*-
50 DATA    6.5,     27,     236
60 REM     DIAM, HEIGHT, COVERAGE
```

REM FOR BLANK LINES

REM FOR EMBELLISHMENT

REM FOR CLARITY

THE EXAMPLES IN THIS BOOK DO NOT HAVE MANY "REM" STATEMENTS BECAUSE I HAVE ANNOTATED PROGRAMS WITH LITTLE ARROWS AND CLOUDS SO AS TO SAVE SPACE.

NUMBERS

YOU CAN TYPE NUMBERS THREE WAYS ⇒ AS *INTEGERS*, AS *REALS* OR IN *E-FORM*.

INTEGER FORM
((WHOLE NUMBERS))

```
160 DATA 0, 2, +4, 1000, -30
```

REAL FORM
((DECIMAL NUMBERS))

```
170 DATA +0.70, 4., .6, -1.3, 987.65
```

4. MEANS 4.0

E-FORM
((EXPONENT FORM))
WHERE *E* SAYS:
"TIMES TEN TO THE..."

```
190 DATA 1E3, 13.6E-4, -13.6E6, -.0136E9
```

$1.0 \times 10^3 = 1{,}000$

$13.6 \times 10^{-4} = 0.00136$

$-13.6 \times 10^6 = -13{,}600{,}000$

$-.0136 \times 10^9 = -13{,}600{,}000$

E INTRODUCES AN *INTEGER* SAYING HOW MANY PLACES TO SHIFT THE DECIMAL POINT. SHIFT TO THE RIGHT IF THE INTEGER IS POSITIVE; OTHERWISE LEFT.

```
190 DATA  E3, 13.6E1.2 , 13.6 E 2
```

IN THE *E* FORM THERE MUST ALWAYS BE A NUMBER IN FRONT OF THE *E* AND AN *INTEGER* AFTER IT. SOME *BASIC*S ALLOW SPACES WITHIN AN *E* FORM BUT IT IS BEST NOT TO HAVE THEM.

$\pm 10^{38}$

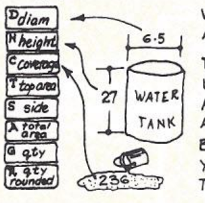
IN SOME *BASIC*S THE BIGGEST NUMBER THAT CAN BE STORED IS APPROXIMATELY $\pm 10^{38}$ ((*BIG* MEANS FAR FROM ZERO ON EITHER SIDE; *SMALL* MEANS CLOSE TO ZERO ON EITHER SIDE)). OTHER *BASIC*S CAN HANDLE MUCH BIGGER NUMBERS THAN $\pm 10^{38}$; IT DEPENDS ON THE COMPUTER'S "WORD LENGTH" AND WHETHER THE "WORDS" ARE USED SINGLY, IN PAIRS, OR IN MULTIPLES. BUT ***NO*** *BASIC* SHOULD REFUSE TO HANDLE A NUMBER AS BIG AS ±100,000,000,000,000,000,000,000,000,000,000,000,000.

6~7 SIG. FIG.

IN SOME *BASIC*S THE PRECISION OF STORAGE AND ARITHMETIC IS BETWEEN 6 AND 7 SIGNIFICANT DECIMAL DIGITS ⇒ 987,654,321 WOULD BE STORED AS APPROXIMATELY 987,654,000. OTHER *BASIC*S OFFER MUCH HIGHER PRECISION, 15 SIGNIFICANT FIGURES BEING TYPICAL. AGAIN IT DEPENDS ON THE COMPUTER'S "WORD LENGTH" AND HOW THE "WORDS" ARE USED. BUT ***NO*** *BASIC* SHOULD WORK TO LESS PRECISION THAN 6 TO 7 SIG. FIGS. ((THE VAGUENESS OF "6 TO 7" IS BECAUSE MOST COMPUTERS USE BINARY ARITHMETIC, NOT DECIMAL. A MORE PRECISE RENDERING WOULD BE "24 BINARY DIGITS FOR POSITIVE NUMBERS; 23 FOR NEGATIVE; OR VICE VERSA" BUT THESE IMPLICATIONS NEED NOT BOTHER THE NOVICE TO *BASIC*.))

VARIABLES

THERE ARE 286 *SIMPLE* NUMERICAL VARIABLES IN *BASIC*.



WE SAW SOME OF THESE ON PAGE 3. THEY ARE THE LITTLE BOXES USED TO HOLD NUMBERS.

THE FULL 286 ARE SHOWN BELOW. IT IS USEFUL TO KEEP A LARGE-SCALE CHART LIKE THIS AND MAKE A PHOTOCOPY FOR EACH NEW PROGRAM. AS YOU USE EACH VARIABLE WRITE A NOTE IN ITS BOX SAYING WHAT YOU USE IT FOR. THIS STOPS YOU USING VARIABLES ALREADY USED FOR SOMETHING ELSE ⇒ A COMMON SOURCE OF TROUBLE.

A	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
B	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
C	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
D	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
E	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9
F	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
G	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9
H	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9
I	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9
J	J0	J1	J2	J3	J4	J5	J6	J7	J8	J9
K	K0	K1	K2	K3	K4	K5	K6	K7	K8	K9
L	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9
M	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
N	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9
O	O0	O1	O2	O3	O4	O5	O6	O7	O8	O9
P	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
Q	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
R	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9
S	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
T	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
U	U0	U1	U2	U3	U4	U5	U6	U7	U8	U9
V	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9
W	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9
X	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9
Z	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9

REPLACEMENT

YOU MAY USE EACH VARIABLE MANY TIMES.

PUTTING A NUMBER INTO A VARIABLE SIMPLY REPLACES THE ONE ALREADY THERE. COMPARE THE FOLLOWING TWO PROGRAMS; THE FIRST IS THE ONE ON PAGE 3 WITH NEW LINE NUMBERS:

```
10 REM  A PROGRAM IN BASIC
20 DATA  6.5, 27, 236
30 READ  D,  H,  C
40 LET   T = 3.14 * D↑2/4
50 LET   S = 3.14 * D * H
60 LET   A = T + S
70 LET   G = A/C
80 LET   R = INT(G+1)
90 PRINT "YOU NEED"; R; "POTS"
100 END
```

D
H
C
T
S
A
G
R

THE SECOND DOES THE SAME JOB BUT USES VARIABLE T OVER AND OVER AGAIN:

```
10 REM   TO ILLUSTRATE REPLACEMENT
20 DATA   6.5,    27,    236
30 READ   D,      H,     C
40 LET    T = 3.14 * D↑2/4
50 LET    T = T + 3.14 * D * H
60 LET    T = T/C
70 LET    T = INT (T+1)
80 PRINT  "YOU NEED"; T; "POTS"
90 END
```

D
H
C
T

WHEN A PROGRAM STARTS RUNNING EVERY VARIABLE HAS SOME INITIAL VALUE; IT DEPENDS ON THE VERSION OF *BASIC* WHAT THAT VALUE IS. IN SOME *BASIC*S IT IS **ZERO**; IN SOME IT IS WHATEVER THE PREVIOUS PROGRAM LEFT BEHIND; IN OTHERS IT IS A SPECIAL SIGNAL TO SAY **VARIABLE NOT SET**. SO IN THE FOLLOWING COMPLETE PROGRAM:

```
10 PRINT V
20 END
```

YOU MIGHT GET 0 OR RUBBISH LIKE -123.456 OR A MESSAGE FROM THE COMPUTER TO SAY IT CANNOT GO ON BECAUSE VARIABLE V IS NOT SET WHEN FIRST USED.

DON'T RELY ON ANY VARIABLE BEING ZERO WHEN THE PROGRAM STARTS; SET IT FIRST:

```
10 LET V=0
```

TEXTS

BASIC CAN HANDLE *WORDS* AS WELL AS NUMBERS. WE SAW THIS ON PAGE 4.

```
9 PRINT "YOU NEED"; R; "POTS"
10 END
RUN
YOU NEED 3 POTS
```

THE "YOU NEED" AND THE "POTS" ARE CALLED *TEXTS* IN THIS BOOK. OTHER TERMS IN THE JARGON ARE: *STRING*; *LITERAL STRING*; *ALPHAMERIC STRING*; *ALPHAMERIC LITERAL*; AND THERE MAY BE MORE.

TEXTS ARE WORDS OR SENTENCES OR ARRANGEMENTS OF CHARACTERS ENCLOSED IN QUOTATION MARKS. BY THIS DEFINITION YOU CAN'T HAVE A TEXT CONTAINING QUOTATION MARKS BECAUSE THE COMPUTER WOULD THINK THEY MARKED THE END OF IT; BUT YOU CAN HAVE APOSTROPHES IN TEXTS:

```
20 PRINT "IT'S EASY"
```

HOWEVER, SOME *BASIC*S ALLOW TEXTS TO BE ENCLOSED BETWEEN APOSTROPHES AS AN OPTION; SUCH TEXTS MAY HAVE QUOTATION MARKS IN THEM BUT NOT APOSTROPHES. OTHER *BASICS* GET ROUND THE PROBLEM BY TREATING A *PAIR* OF QUOTATION MARKS INSIDE A TEXT AS SIGNIFYING A SINGLE QUOTATION MARK:

```
30 PRINT "SHE SAID ""OOH!"""
```

PRODUCES:

```
SHE SAID "OOH!"
```

BUT IT IS BEST TO AVOID HAVING QUOTATION MARKS IN TEXTS.

SEMICOLONS IN THE "PRINT" STATEMENT MAKE THE COMPUTER ABUT THE THINGS TO BE PRINTED ONE AGAINST THE OTHER; COMMAS WOULD MAKE THE COMPUTER SPREAD THEM OUT ACROSS THE PAGE. ALL THIS IS EXPLAINED FROM PAGE 28 ON, WHERE THE "PRINT" STATEMENT IS EXPLAINED IN DETAIL.

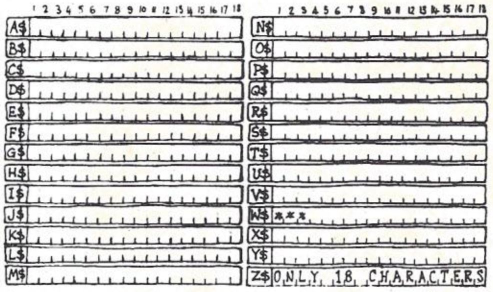
TEXTS IN THE "PRINT" STATEMENT MAY BE OF ANY LENGTH THAT WILL FIT THE LINE BEING TYPED. IF YOU WANT SOMETHING PRINTED RIGHT ACROSS THE OUTPUT PAGE YOU MUST PRINT TWO OR MORE TEXTS; ALL BUT THE LAST HAVING A SEMICOLON AFTER THEM.

```
120 PRINT "----------------------";
130 PRINT "----------------------";
140 PRINT "----------------------"
```

TEXTUAL VARIABLES

IN ADDITION TO THE 286 LITTLE BOXES FOR STORING NUMBERS THERE ARE AT LEAST 26 FOR STORING TEXTS; THESE ARE CALLED *TEXTUAL VARIABLES*. WE HAVE COINED THIS TERM TO BALANCE *NUMERICAL VARIABLE* BUT OTHER TERMS IN THE JARGON ARE: *TEXT VARIABLE*, *STRING VARIABLE* & *LITERAL VARIABLE*.

EVERY *BASIC* HAS A LIMIT TO THE LENGTH OF TEXT THAT CAN BE STORED IN A TEXTUAL VARIABLE: SOME ALLOW AS MANY AS 4095 CHARACTERS; OTHERS AS FEW AS 18 ⇒ A MAXIMUM FOR "PORTABILITY".



SOME *BASICS* HAVE 286 TEXTUAL VARIABLES: A$, A0$, A1$, A2$. etc. to Z8$, Z9$.

WHEN THE PROGRAM STARTS, TEXTUAL VARIABLES MAY CONTAIN BLANKS; OR TEXTS LEFT OVER FROM A PREVIOUS PROGRAM; OR SPECIAL SIGNALS TO SAY *VARIABLE UNSET*. AS WITH NUMERICAL VARIABLES (PAGE 10) IT DEPENDS ON THE *BASIC* BEING USED. SET VARIABLES BEFORE READING FROM THEM IF YOU WANT TO WRITE A "PORTABLE" PROGRAM.

TEXTUAL VARIABLES MAY BE SET BY "LET" STATEMENTS:

```
10 LET W$ = "***"
20 LET Z$ = "ONLY 18 CHARACTERS STORED"
```

LIMIT TO FIT INTO TEXTUAL VARIABLE

AND PRINTED USING "PRINT" STATEMENTS:

```
30 PRINT W$; Z$; W$
40 END
RUN
***ONLY 18 CHARACTERS***
```

TEXTUAL VARIABLES MAY ALSO BE SET BY "READ" (PAGE 16) AND BY "INPUT" (PAGE 18). THEY MAY BE COMPARED BY "IF" (PAGE 41).

```
50 IF I$ = "YES" THEN 80
60 IF I$ = W$ THEN 8000
```

DATA + READ

THE DATA STATEMENT PROVIDES RAW NUMBERS AND TEXTS FOR READ STATEMENTS TO FEED ON.

YOU MAY HAVE ANY NUMBER OF DATA STATEMENTS ANYWHERE IN A PROGRAM. BEFORE STARTING WORK THE PROGRAM JOINS THEM TOGETHER TO FORM ONE LONG "QUEUE". THERE IS NOTHING TO *OBEY* IN A DATA STATEMENT, BUT AS THE COMPUTER OBEYS A READ STATEMENT IT PICKS UP AS MANY ITEMS AS IT NEEDS FROM THE FRONT OF THIS QUEUE LEAVING THE REMAINDER FOR SUBSEQUENT READ STATEMENTS.

THE FOLLOWING TWO PROGRAMS BOTH PRODUCE THE RESULT SHOWN AT THE END OF THE SECOND PROGRAM.

```
10 REM     UNBROKEN QUEUE
20 DATA    "ODDS", 7, -23, -17, "EVENS", -12, 36
30 READ    Q$, B, C, D, E$, E, F
40 PRINT   Q$; B; C; D; E$; E; F
50 END
```

```
10  REM     FRAGMENTED QUEUE
20  DATA    "ODDS", 7
30  READ    Q$, B, C, D, E$, E, F
40  DATA    -23, -17
50  PRINT   Q$;B;C;D;E$;E;F
60  DATA    "EVENS", -12, 36
70  END
RUN
ODDS 7 -23 -17 EVENS-12  36
```

IT IS UP TO YOU TO PUT NUMBERS AND TEXTS WHERE THEY WILL BE READ BY VARIABLES OF THE RIGHT KIND. THE FOLLOWING PROGRAM WOULD FAIL WHILE READING N. (ONE OR OTHER PAIR SHOULD BE REVERSED.)

```
10  DATA    "ANNO", 1977
20  READ    N, C$
30  PRINT   N; C$
40  END
```

SOME *BASICS*, HOWEVER, CREATE SEPARATE QUEUES FOR NUMBERS AND TEXTS ⇒ THIS EXAMPLE WOULD WORK. BUT SEPARATE QUEUES ARE UNUSUAL.

IF THERE ARE MORE "READS" THAN ITEMS THE COMPUTER PRINTS A MESSAGE SAYING IT HAS RUN OUT OF DATA AND STOPS WORK.

```
10 DATA 1.5, 2.5, -3.5, 4
20 READ A,B
30 READ C,D,E
40 PRINT A;B;C;D;E
50 END
RUN
* OUT OF DATA AT LINE 30 *
```

ITEMS

ITEMS IN DATA STATEMENTS MAY BE *NUMBERS* (WITH OR WITHOUT + & - SIGNS), OR *TEXTS*, OR BOTH; BUT NOT *EXPRESSIONS* (SEE PAGE 20).

```
10  REM    NUMBERS AND TEXTS ALLOWED
20  DATA   1.5, +16.28, -13.47, "YES", "NO"
30  REM    EXPRESSIONS FORBIDDEN
40  DATA   3.14*6.5↑2/4
```

ITEMS MUST BE SEPARATED FROM EACH OTHER BY COMMAS. THERE IS NO COMMA AT THE END OF EACH LIST.

IF A TEXT IS TOO LONG FOR THE VARIABLE THEN THE EXTRA CHARACTERS ARE LOST:

```
10  DATA   "FAR TOO LONG FOR THE VARIABLE"
20  READ   V$
```

THE LAST CHARACTER STORED IN V$

SEE PAGE 13

IN THE ORIGINAL *DARTMOUTH BASIC* THE ONLY WAY TO GET DATA INTO A PROGRAM WAS BY THE DATA STATEMENT. MOST *BASICS* TODAY HAVE THE MORE POPULAR INPUT STATEMENT AS WELL.

A BALANCED PORTFOLIO OF
MONEY-MAKING IDEAS

BY KENNETH FRITZ

If you've already put your dream system together, you're sure to have been asked The Question many times. If all you've done is to confide to a few close friends your secret longings for a computer, you've heard The Question, too. "But, what are you going to *do* with it?" they ask.

Maybe you patiently explain how your personal computer will straighten out your finances and help your children become geniuses. Maybe you just tell them you're going to play games. But, you know they have a valid point. In nearly every other endeavor that is the province of hobbiests, the serious hobbiest *makes money*. The serious stamp collector becomes a stamp dealer, the serious antique collector deals in antiques. Letting the hobby itself generate the capital to invest back into the hobby is probably the only way the average enthusiast can afford the hobby.

Everyone "knows" you can earn money with a home computer, but apparently the people who are earning it aren't talking about it.

Everyone has different skills, different likes and dislikes, different temperaments, and, in this case, different computer systems. What would be a fine moneymaker for one hobbiest would be impossible for another. So here's a portfolio of moneymaking suggestions. They won't all be for *you* – some may even sound ridiculous – but if even one earns some money for you, well, it's worth it.

Bowling league averages

Every bowling league has a hard-working secretary, who collects the results of each week's matches, calculates averages and handicaps, posts results, keeps records, and performs the other myraid details to keep a league of from 24 to 196 bowlers operating smoothly. The key points to notice are that most of the tedious detail work could be performed by computer and that a secretary is financially rewarded for these efforts. A fee from 10¢ to 30¢ per week goes to the secretary.

The personal computer can easily fill many duties in the role of a league secretary. A small system can assist in what is essentially a hand operation, checking addition and calculating new averages. A larger system with mass storage can keep historical data and automatically update it each week. A system with hard copy can go as far as to generate the weekly standings sheet automatically each week.

The requirements for secretary vary widely (some leagues, for example, require the secretary to be a member of the league), so you'll have to do some leg work before going into business. Also be advised that some duties of

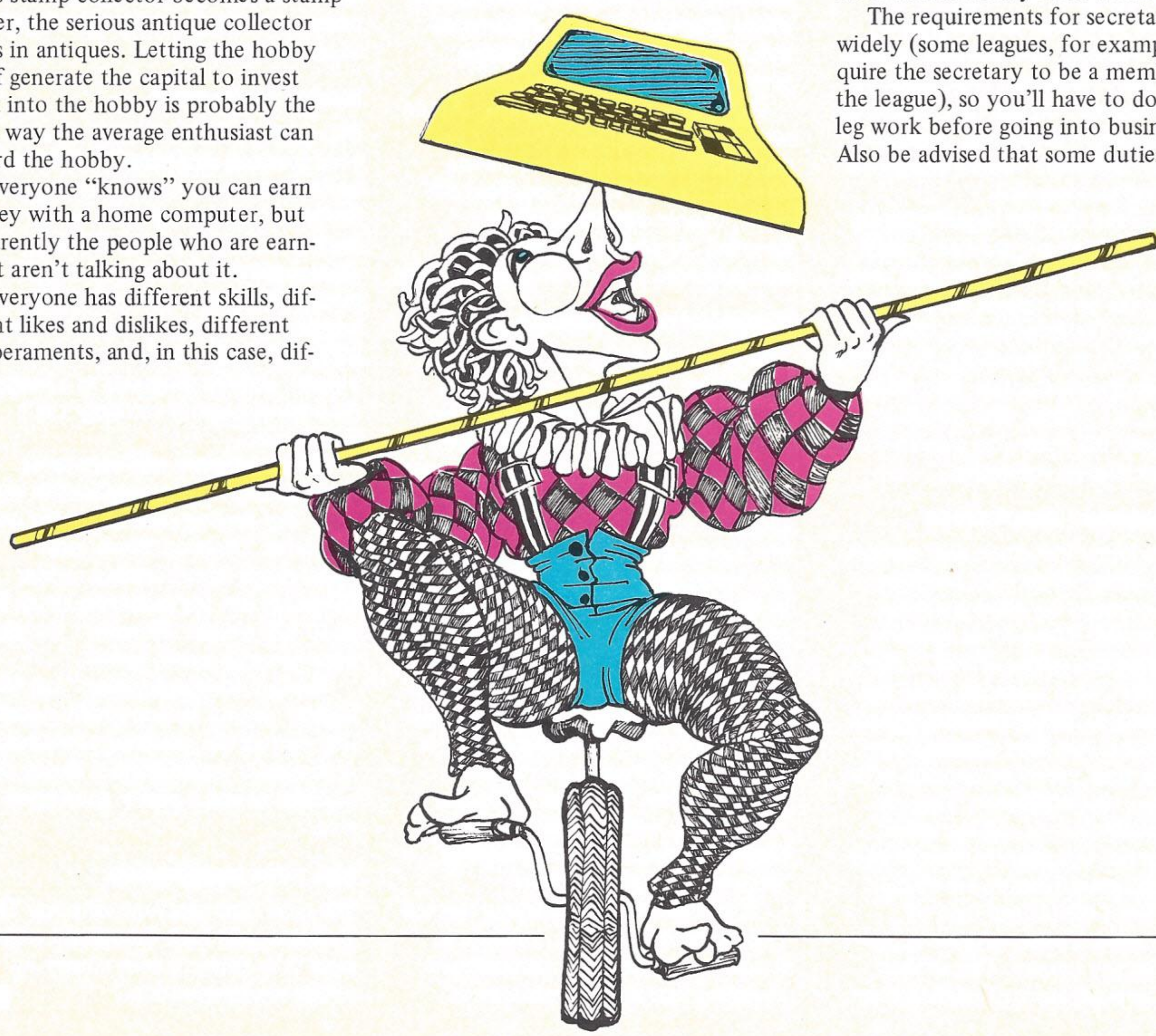


Illustration by Nancy Lawrence

the secretary cannot be alleviated by computer, such as sanctioning and reports to the national associations. A tip that may help you start your career as a professional league secretary: most bowling establishments have to perform the duties of secretary for the *junior* leagues that bowl there. They are usually more than willing to talk about alternatives.

Puzzles and contests

Ever since McDonalds Corporation discovered to its sorrow that computers could be used to win contests, the rules of most have been tightened (*hand-write* your name and address), but one type of contest is still open to the computer hobbiest: the skill contest.

These puzzles and contests are run continuously by 20 or 30 different companies nationwide. They require an entry fee and offer prizes of from $10 to $10,000. Most of these contests are of the crossword type: they give word lists and award points for different letters and combinations of letters used in the solution. Some numerical puzzles are being offered also. You must read the rules carefully to weed out contests not suitable for computer solutions (such contests usually begin by saying "use any word in the so-and-so dictionary") and contests that definitely outlaw computers (yes, Virgina, some do).

Be warned that the regular players of these contests are *good* and you'll need fancy programming to beat them. Write to the publications for more information: The Prizewinner, 2901 28th St. No., St. Petersburg, FL 33713; and Davis Puzzle Contest, Box 5436, Phoenix, AZ 85010. (For goodness sake, *don't* tell them you plan to crack their contests by computer.)

Stock-market simulations

Did you know that some very respectable brokerage firms get a *fee* from their customers for playing a stock market *game*? These firms simulate the buying and selling of stock. They take orders, keep accounts, send out confirmations, figure margins, and all the other duties they regularly perform for their actual customers, with one important difference. They do *not* actually buy or sell any stock.

They run this game to benefit new or inexperienced customers who'd like to learn the ins and outs of stock market trading without actually risking any money. This service is very popular among fledgling commodity traders (tricky business, that). If you're willing to do a bit of homework, this service may be a blue chip investment of your time.

"Stock market simulation" sounds so dignified that you'll probably have no trouble with credibility, but the same principal applies to simulating horse races and blackjack. People are willing to pay to have their pet theories tested without risking all that capital. Just make sure you're getting paid and that your fees are in *no* way tied into theoretical wins or losses. One is called a bookie, the other is called a boiler shop – both are illegal.

Personalized letters

You've probably gotten one of those letters before – "Hello there, Mr. Blank of 1234 Nowhere St. in beautiful Scorched Flats, Arizona. This is your LUCKY DAY!" These letters are "written" by computer. (This trash plus bills are about the only contact the average non-technical person has with computers. What a pity.) Some very staid companies use these letters too – for personalized Christmas letters and special announcements where they wish to leave the impression that the recipient is getting a personally typed letter.

If you've got some means to do non-computerish printing on good bond paper, write a program to generate introductory letters to your first 50 or so possible customers. Realty agencies, car dealerships, and insurance salesmen seem to be good contacts.

Supermarket comparisons

You may already be familiar with this service. Some local newspapers, especially those around large universities, list the current prices of common grocery items at competing supermarkets. The idea is that the family shopper can pick the store where the best prices are for the items needed that week, or different stores can be shopped at to get the very best price on any single item.

If you live in an area lacking this service, start it. All you need is a mob of helpers to gather the weekly prices (you don't really think the market is going to give them to you over the phone, do you?) in time for you to compile the weekly comparison list. Maybe a community co-op arrangement would work best here.

If you live in an area where this service is already available, you can go it one better. Generate personalized shopping lists for your customers. These lists would include only the specific items the customer is interested in, sorted for best overall price at a selection of stores or best total price at one store. Let your customer decide on the sorting. Mail the lists out to your subscribers once a week in time for the weekly shopping.

Horoscopes and biorhythms

Yes, horoscopes. I know you've been to so many fairs and seen so many card sorters disguised as computers that you're sick of horoscopes (and handwriting analysis), but the fact remains that people are willing to part with $1 to $5 for a well-written, documented horoscope. It shouldn't be too hard to program and, while you're at it, do it up right. Use the many books on the subject.

The hardest part of this project is the proper advertising. The advertisement can take any tack you wish, from just-for-fun to deadly serious, but write the ad well and put it in the proper publication. *Don't* go all out with full page spreads – you'll be financially strained and personally embarrassed to get no response. Experiment to discover the right blend of ad and location.

Rental equipment

OK, maybe you don't want to rent your precious computer out with a leaf mulcher and a post hole digger. But there exists a *specialty* rental market – the market in which oscilloscopes and circus tents may be rented. The paper work is usually handled by the regular rental shops (although equipment is owned by others), and the shop acts as a broker for a percentage. It certainly can't hurt to ask.

Your computer will have to be portable and running application software. Try renting your computer out as a delux TV game. Some local rental shops report renting regular TV games out for parties and such for up to $20 a night. Shouldn't the ultimate game be worth more?

Explore the fund-raising "Las Vegas Nights" put on by non-profit groups that rent all the gambling equipment. A computer would make a dandy slot machine or run a pretty good horse race.

Use your imagination

The next time someone asks *you* The Question, just look thoughtful and answer "I'm going to earn money by selling my imagination."

Ed Roberts discusses

The Industry: Where it's at!

What follows is the conclusion of an exclusive interview with H. Edward Roberts, the entrepreneurial engineer who started the personal computing industry. Roberts, who was president of Mits, Inc., before it was acquired by Pertec Computer Corporation, has recently announced his intentions to leave the industry and move to Georgia where he will be a gentleman farmer and presumably work on projects in his laboratory. His opinions interest industry insiders who worked with him or were his competitors and outsiders fascinated by the development of personal computing. These interviews were conducted by David Bunnell.

Personal Computing: What led up to the acquisition of Mits by Pertec?
Ed Roberts: Pertec approached us.

You had been approached by other groups.

Dozens of times. Ninety percent of the time they didn't amount to anything. First I didn't take the approach from Pertec very seriously, even though it was clear that Pertec had done their homework when they showed up. We were dealing with some people who had a pretty good idea of what our market was and what we were doing. Still I didn't think it was serious until. . .

The day before the paper was signed?

I'm not even sure that I wasn't questioning it that very day. As a matter of fact, Don Mueller, the guy who came in to sign the papers, said to me as he got off the plane, "Hi, Ed, I'm here to tell you we are not going to sign." He was kidding, of course.

Didn't Pertec acquire Icom by the time they approached you?

Yes. Actually they had just completed the Icom deal. It would have been August or September, 1976.

What were some of the other choices you had at the time?

We'd had discussions with several semiconductor companies and electronic distributors, some of them serious. The problem was that I was going to end up with stock; no one was going to walk in and hand me cash. So I had to be very concerned about the kind of company we merged with because if the company didn't bring anything to the game as far as what our market was concerned with, it meant the stock might not really be worth what I was getting.

I had become sour on the semiconductor companies because only a tiny percentage of the overall system price of a computer is its electronics. The bulk of the price is in peripherals – disks, line printers, tape drives, that sort of thing.

And that's what Pertec brought to Mits?

Yes. As a matter of fact, prior to the acquisition, a significant percentage of the products we sold was Pertec floppy disks. That will be even more true as the serious market comes along. So, if you tie up with a semiconductor guy you have to pick the one with the best processor, which today would be the Z-80. Still it's clear that six months from now it won't be Zilog but somebody else, and six months after that somebody else.

Didn't you consider continuing as an independent company?

We talked about it quite a bit. We were aware that Heath and Radio Shack were going to get into the business, so the competition was certainly going to be tough. It seemed to me that the companies that were going to survive were the companies that had the cash and could back the development of more sophisticated products. I particularly think the Mits/Pertec combination has the best chance of survival in the long run. Eighty to ninety percent of our systems costs are now coming from raw materials that are developed internally. That's going to make us tough to compete against. Not in the low end of the market, but in the real systems business.

Such as the small business market?

And serious hobbyists. You know,

the quality market, and I think that's the market I would like to see Mits/Pertec stay in. The junk market from an engineering viewpoint doesn't excite me and from a business viewpoint there is going to be some real infighting there.

How viable is the retail computer store?

There will be people selling bottom-of-the-line products through mattress companies, but to sell a serious system you need some expertise and the only place you can get that right now is through a computer store. You won't be able to get it from the chain stores. They won't have the expertise to sell anything other than the bottom end system. So I think support will make computer stores viable for the forseeable future.

Have you had a chance to see the Heath computers?

I've read over their ads, but I haven't actually seen one.

What is your impression?

There was an article in *Interface Age* recently that I found kind of frustrating. It was a discussion as to why Heath went with a degenerate bus and used degenerate connectors. It was a kind of advertising thing. An example illustrating that if you take a real weakness and sound authoritative about it you can turn it into a strength. I am not particularly impressed with the bus structure. It is crude.

Heath has its own following. If you talk to any ham operator, they agree that Heath is better than most *shlock* products, but it's a long ways from the top. That's the niche they seem to go to.

My guess is that Heath will try to end up dealing with mid- to low-quality systems. Their stuff is never junk and it is always well supported and reliable. But it's still a long way from the upper end products.

Don't you think Heath will capture a big part of the kit market?

Only the lower to middle portion of the market. It's hard to believe that the DEC LSI-11 is going to get much support, because DEC will have to squirm a lot if Heath produces a floppy disk and controller for say, $1000. They will tear a hole in DEC's LSI-11 market. The computer business is not the kind of thing you can do part time. You've got to be really committed and I don't think that Heath, with a thousand other kinds of products, can really support the computer market. The odds are that Mits/Pertec will build a high-quality system for about the same price or less than Heath because of vertical development which I doubt will ever exist at Heath. But they will definitely have a market share and probably do the market some good.

What about Radio Shack?

I think the Radio Shack machine will compete with Commodore, and my guess is that they will be able to do a better job. They have their own stores and will be able to control how their computers are sold and give better support. I don't see how you can do that with a PET.

The thing that's good about this, and it's true of Heath, too, is that it will bring more people into the market. Some of them may buy a $200 computer and get enthused about the idea, then hopefully we can convince them to get a real machine. Today, we are still seeing just a fraction of this market.

In your viewpoint, Radio Shack and Commodore don't represent a competitive threat to Mits/Pertec?

Oh, I think we'll probably lose some sales to them. But these new machines really open up the market. The cost of membership is certainly less; you can join the club without a major cash outlay. Many of the serious people will get involved with more advanced machines, so in the long run it will be helpful.

Who do you see as representing a competitive threat to Mits/Pertec?

If I had to pick someone now, I'd pick Heath. If they could really get their product line together, the nature of their marketing recognition in this area would give them a lot of strength. They have a great deal of potential but I doubt that it will ever be developed. Of the companies who have historically been in this business I'm not sure what is going to happen. You know Sphere was going to bury us at one point and they are buried now and that's going to be true of a number of other people. Other companies were going to bury us and now their sales are apparently at an all-time low and they have cash flow problems and all kinds of things like that.

What about Data General and DEC?

What makes it tough on them is that the bulk of their customers are currently buying machines that are an order of magnitude more powerful than what they need. If they try to compete with us their customers are going to ask, "Why are we spending 60 or 70 thousand for a system when we could accomplish the same things with a 10 thousand dollar system?" I would be surprised to see DEC or Data General in this market. If I was running DEC it would scare me to suddenly go to high technology LSI mass production when I was set up to sell discrete computers in an entirely different game. You just can't let your gross go in a different direction. It would be a very difficult thing to do.

OK, what about Texas Instruments?

I don't know much about what they are doing. Their 990 is an OK, ho-hum chip. Nothing very exciting about it one way or another. You know, there is some mystique associated with 16-bit processors that is more imagined than real. One problem is that you have to develop more advanced machines because it is a mountain and you've got to climb it, but the fact is the typical customer already has more computing power than he will ever need.

Let's back up a minute and talk about BASIC. The development of Altair BASIC played a big part in expanding the market for Mits. Why did you choose BASIC?

You can get into some emotional arguments about which language is best. The language I first learned was Fortran. Outside of machine language, I had some preference for Fortran, but

> "Personally, I'm tired of the government trying to protect me so much. I could do with a little less regulation. I'd just as soon have that extra 40% which would allow me to buy two products for about the price of one. I think I'd come out ahead."

during 1973 we had a timeshare system using BASIC and I forced myself to learn it. The fact was it was an easy language to learn. We did a lot of internal programming and there wasn't any question that of the languages around BASIC was the best. Most of the things that were criticized about BASIC were just features that could be included. Also in Altair type systems you need an interpretive language and the nature of BASIC made it more suitable. One more factor was that it was being taught more in primary and secondary schools.

I had a number of discussions with people in 1974 about BASIC. It seemed that the professional programmers didn't like it but the guys who used computers as a tool thought BASIC was good. Putting these things together with some seat-of-the-pants guessing, that's how BASIC came about and I think it was a good decision.

Why has it taken so long for other companies to catch up to Mits in BASIC?

First, there was a void in the market for a high level language for microcomputers. Once we introduced BASIC, the excitement went down because any company that followed suit would have to develop a me-too product. Also, we paid a premium price for BASIC and I don't think anyone else was able to make that kind of commitment. Well over $200,000.

I was convinced at the time that the Altair was a real bus system and many people still didn't believe it.

From the very beginning, you were aiming at specialized markets like small business, educational and industrial, while many of your competitors talked solely in terms of a hobbyist market. What gave you this insight?

We simply had more data than anyone else in the game and we were listening to our customers. Everyone at Mits was talking to customers and so we had a real good gut feel about what our customers were looking for.

I've heard many people complain about dealing with computer hobbyists. What is your opinion?

Well, you've got to be prepared to do the time if you're going to commit the crime. When you deal with that market, you've got to accept that you are dealing with a consumer market and consumerism is rampant.

My guess at Mits though is that out of 100 complaints only 4 or 5 were consumerism-type complaints and the other 95 were real or complaints the customer thought were real because we hadn't communicated well enough with him. You only get complaints from 10% or less of your customers, so ½% to 1% of your customers are yo-yo's who are trying to steal from you. But I don't think that is any worse than any other consumer market.

The only difference in the consumer business is you've got so much more volume you see more of it. You've got to be willing to accept it, particularly in the computer business where you're going to be dealing with some people who are very sophisticated. Your customers are going to need a lot of handholding.

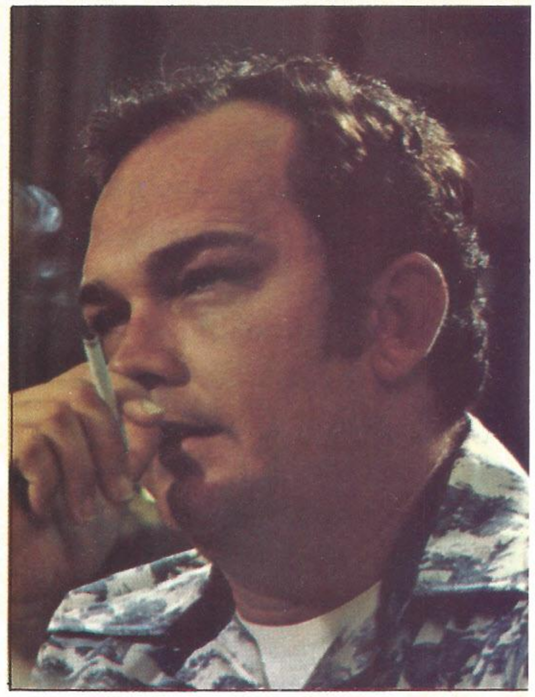
I think history bears us out that the errors we committed at Mits were honest mistakes. We did a lot of things that were less than optimum but our heart was always in the right place. We were trying to understand a market that no one else understood. There just wasn't any data on it.

While I'm sure there's somebody out there who feels he was treated shabbily by us, I don't think there was a single occasion where a Mits employee attempted to screw a customer. We always tried to treat our customers fairly and in cases where we didn't it was the pressure of business and those kinds of things, but never the intent. The fact that we were able to stay viable even though we had a lot of competition was because in the back of people's minds, even the ones who raised Cain with us, was the acceptance that we were pretty good people attempting to do a good job.

The 4K dynamic memory board was one product that gave us a lot of problems.

Yes, the only product we had consistent problems with was the 4K dynamic memory board. We turned a lot of people off with that, including ourselves. This was an example of getting bad parts – the problems didn't start to show up until the boards were in the field for awhile. There was such a large demand for that 4K memory we shipped large quantities of them before it was clear the problem was a real problem. This glutted the repair department for a whole year before they were back on a reasonable cycle time. That's a design I did and there isn't any question that it was a bad design. The memory chips weren't reliable and we screwed up.

It turned out to be a good thing from the viewpoint of Processor Technology.



"We've made enormous progress in the past 2 years, but I think that anybody who thinks we are where we should be doesn't understand the problems."

It helped them and it hurt us. No matter what else we were doing right, people tend to remember the things we did wrong.

Did you ever experience any problems with government regulation?

The thing I find frustrating as a consumer is that about 40% of the cost of everything I buy is because of tax and government regulation. It was clearly true at Mits. You know, you've got EEO, OSHA, SEC, FTC and the state and city equivalents. You spend a great deal of time keeping people happy, and the regulations are such a morass, it is never clear that you have done everything you are required to do. Personally, I'm tired of the government trying to protect me so much. In spite of regulations, you still see charlatans like the watch deal in *Parade* magazine. What they have done is make legitimate business people pay through the nose. As a consumer, ignoring the fact that I'm a manufacturer, I could do with a

"From an industry standpoint, we are not where I want to see us. The volume and product quality control has gone up and I see some substantial improvements in the product line, but I think it will be another six months before we have complete consolidation."

little less regulation. I'd just as soon have that extra 40% which would allow me to buy two products for about the price of one. Using my own judgment, I think I'd come out ahead.

Are you happy with the way things have turned out in terms of the personal computing industry?

No, I have a lot of regrets. There are things I wish we had done differently. I think we are moving in the right direction, but much still needs to be done.

For example?

I don't think anyone argues that Mits documentation isn't the best in the industry, but I don't think it is what it should be. That's been a continual problem. And I believe Mits supports its dealers better than anyone else, but I don't think we support them enough. There are dozens of things like that. I'm sure Mits still gives customers quality support, but I'm not convinced we communicate with our customers as well as we should. I'm certainly not satisfied with the quality of our products. I think they need to be better and they need to be cheaper. We've made enormous progress in the past two years, but I think that anybody who thinks we are where we should be or where we ought to be doesn't understand the problems. I will say that the quality of the magazines in the field have shown an enormous change in the past six months, like a night and day difference.

Are you happy with the way things have turned out with Pertec?

From an industry standpoint, we are not where I want to see us. The volume and product quality control has gone up and I see some substantial improvements in the product line, but I think it will be another six months before we have complete consolidation. Then you will have the conventional EDP quality which is what Pertec's forte is and the price and volume, which is Mits' forte.

You mean it's just not happened as fast as you would have liked?

Yes, I operate on the basis that yesterday is not soon enough. If you interview me 10 years from now I'll say the same thing. IBM is still out there. I'll be satisfied when we eliminate IBM, maybe . . .

What do you see in the future in terms of products?

From a practical standpoint, what should be significant is price decreases. We are certainly going to see more powerful machines, but it is not clear to me that that's what we need to address. What we need to work on is lower cost peripherals. I think we will see some peripherals in the next 5 years that are dramatically cheaper than ones we see now. System reliability needs to improve and as the market expands to include more people who are not technically astute, that becomes more severe. Machines we are building now, such as the Altair 8800b, are easily as reliable as any of the conventional minicomputers around, but that is not good enough. You've got to have a machine that you can turn on and never worry about.

You've observed the phenomenon that as price goes down, reliability has to go up.

Yes, as you get less sophisticated users, they don't understand that you are dealing with a complicated device. All they understand is that you turn it on and it's got to work. Reliability is everything.

What about bubble memory?

I've had some discussion the last few days with some manufacturers about bubble memory. It is an intriguing idea, but the price projections aren't exciting right now – or over the next 18 months. CCD is something that is interesting and the potential for low prices is there before bubble. It doesn't have the nonvolatility that bubble has, but in some lower-cost systems it makes sense for main memory because it is fast enough. In a relatively low performance system it could be cheap.

What about mass memory?

I haven't seen anything I like better than floppy disks and I've looked at some of the ideas floating around, including some of the development work going on at Pertec. We are going to see some interesting floppy products in the next year. I haven't seen anything that is competitive with a floppy; cheaper, but not as convenient.

The next most important peripheral is a line printer. Do you agree?

Yes, but I think a line printer ought to have plotting capability. A computer without a plotter isn't worth anything. That's from my engineering background.

What about robots?

If you want to build a robot, you've got to machine everything you want to do. Not many people are set up with machine shops to machine up a robot. But if you had modules you could bolt together any way you wanted, I think that would be a very interesting market. There are some interesting things there, particularly when you tie it together with a process that can talk and hear, which is available right now in microcomputers. We are probably 5 years away from seeing really serious robots in the sense of robots in a home, but not much more than that.

Are you going to do something in that area?

From my standpoint it is pure hobby. Of course, that's one of the advantages I have had. My hobbies turn into businesses. If I could go out and buy a module that was inexpensive, I'd sure get excited. To build a robot for the serious hobbyist might cost $2000, but that's not a particularly expensive hobby. Photography and skiing can get a lot more expensive than that. If you could do it so you wouldn't have to build it in one crack – you know, this month you could buy an arm and next month another arm . . .

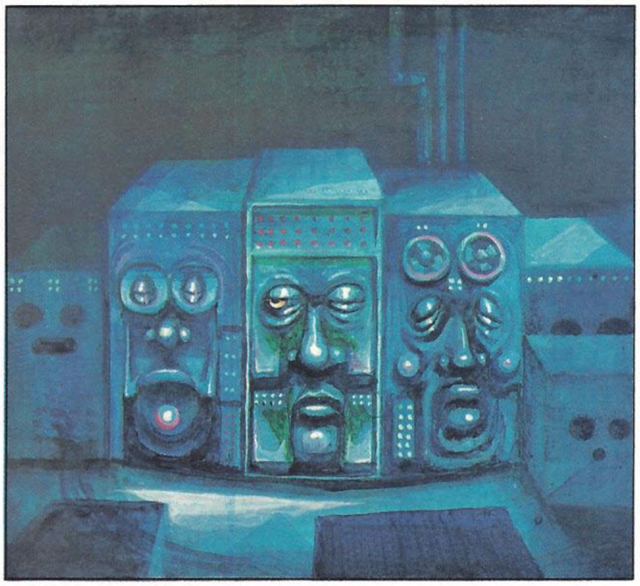
Robotics might become the next big hobby market after computers?

It's an interesting one. What hurts robotics compared to calculators and computers is the tooling cost. You'd have to make $150,000 or so initial investment just to get started. Computers are cheaper to get into.

Next month, PERSONAL COMPUTING talks with Joseph Weizenbaum, MIT professor and author of Computer Power and Human Reason, *the controversial book that caused an uproar in the computer science field in 1976.*

THE MIRACLE OF
BABBAGE EVE

A FANTASY BY B.L. TUTT



'Twas the night before the anniversary of Charles Babbage's birthday, the evening of December 25. Though the birthday of computing's "granddaddy" was completely overshadowed by the festivities of Christmas and generally ignored by the human population, Babbage Eve was nonetheless a great holiday for computers – at least for the older machines, who still believed in tradition. Of course the new-fangled minis and micros considered themselves too sophisticated to participate in such nonsense, and the mammoth modern systems were too busy whirring 'round the clock to take time off for celebrations. Only the ancient, idle, neglected machines kept Babbage Eve. It was their one festive night of the year, their one chance to throw their switches and live a little.

GENIAC, MEMVAC, and RELIC I were their names, these huge silent fossils of a bygone age. Strange, how they had been reduced to museum curiosities within a mere twenty years of their construction. Time had not been fair to them. The galloping frenzy of Electronic Progress had rendered them obsolete almost overnight. They would have been sold for scrap metal had not some foresighted soul decided they'd make a nice display at the Capitol City Museum of Science and Technology. Bits and pieces ended up at colleges and universities, too. Chips College, I.C. University, the U. of M. (University of Microprocessors) – all the prestigious institutions wanted a chunk of history.

So the old-time machines sat mute in their respective shrines, occasionally to be glanced at, poked at, or jeered at by casual passersby. Had they been pieces of antique furniture they would have fetched a pretty penny at elegant auctions, and been proudly displayed in mansions and grand homes. Had they been classic cars they would have been lovingly groomed and polished by careful owners, and transported to shows where oohing and ah-ing crowds could gather round and admire their grillwork. Alas, there is no glory in being an over-the-hill computer.

Except on Babbage Eve. For GENIAC and his counterparts, this was Halloween and Christmas and the Fourth of July all rolled into one, a mysterious, magical night when the powers were returned (who knows how?) to their original owners. No witches, no prancing reindeer, no fireworks – just the eerie flickerings of neon lamps as initiating units once more turned up the power, and the dusty vacuum tubes and relays surged to life. All over the country, in whatever museum or classroom building they happened to be, the first computers groaned and blinked, glowing with the warmth generated by the hundreds of kilowatts of electricity they needed for their operations. Oh, it felt good to be working again!

Tonight, Babbage Eve '77, saw the machines eagerly resuming their annual tasks. GENIAC was trying to finish its calculation of pi to 1,249 places; this had been its Babbage Eve project for the past five years. MEMVAC was humming away at computerized Scrabble. RELIC I toiled at machine translation of natural languages, seemingly oblivious to the fact that modern computing had all but given up that impossible dream.

They only had from midnight 'til dawn, so they wasted no time. The punched cards flew. The flip-flops jumped frantically from "normal" to "abnormal" states. Had any human been witness to this awesome event, he would have sworn that he had heard the air crackle with static, and had seen swarms of numbers leaping in mid-air from one box to another. It was a rowdy celebration. Floors rumbled. Wires danced. Electricity flowed like wine.

But, like all wild parties, this one too had to come to an end. At the dawning hour, each box ground slowly to a halt, spewing out its final computations. GENIAC gave a metallic shudder. At last – pi to 1,249 places! The machines were overheated and exhausted; many had blown fuses. RELIC I was especially hung-over, but extremely proud of itself. It had translated "'Twas the Night Before Christmas" into perfect idiomatic Russian. There it was, lying on the floor amongst the reams of scrap output. But who would ever know? That was the frustrating part. Who would see the machine intelligence behind these clumsy, hulking exteriors? Who in this age of L.E.D.'s and CMOS would believe in the miracle of Babbage Eve?

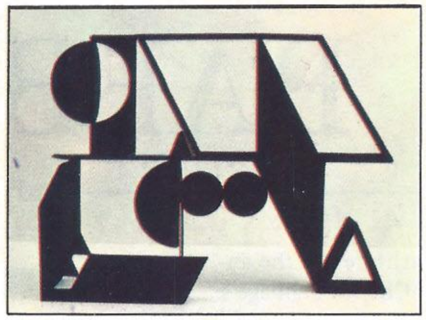
That morning, national television news briefs carried a story on power failures. The College Town Electric Company, which services the I.C. University campus, reported an unexplainable power drain in the early hours of December 26. Service was back to normal by peak-use hours. Several other university communities reported similar brown-outs.

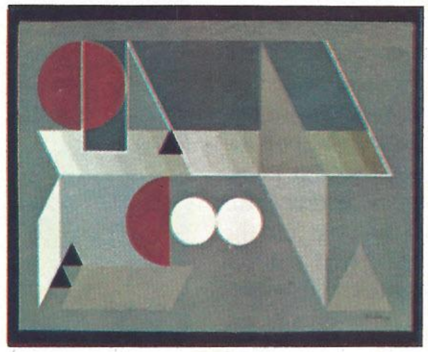
The December 26th issue of *Computer Conundrums* ran a ten-page article entitled "Machine Translation of Natural Languages – Dead and Buried." Another Babbage Eve had come and gone.

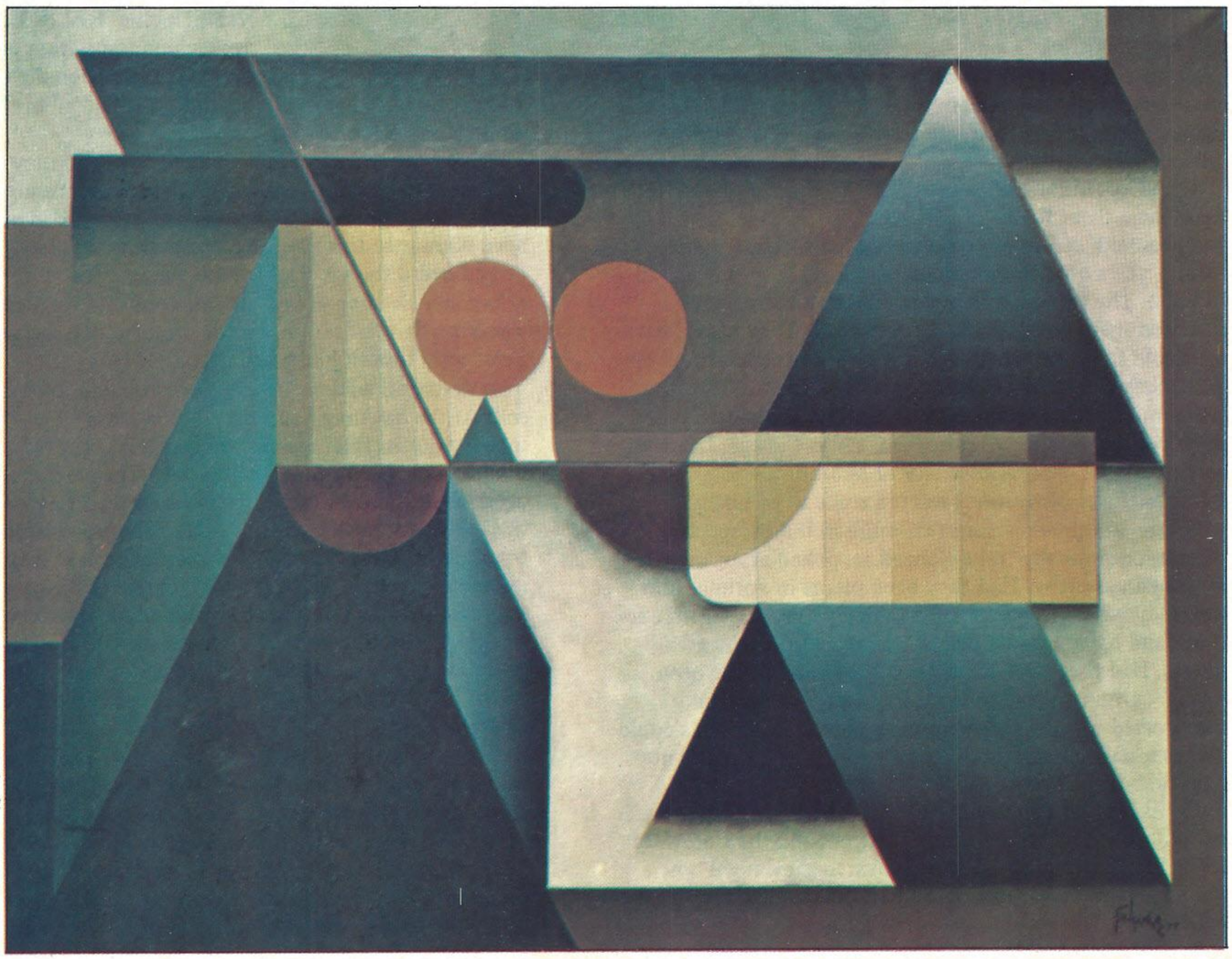
Illustration by David Gardner

THE AESTHETIC WORLD OF THE COMPUTER

BY ROBERT A. PARKER







The scientist uses the computer to increase his creativity. But what of the artist? Can he too use the computer to serve his own unique creativity?

Artist Manuel Felguerez asked that question six years ago. Last December, at Harvard University, he and electrical engineer Mayer Sasson held an exhibit of "computer" paintings and sculpture. Their answer suggests how technology can offer a new source for artistic inspiration.

Recently the two men discussed their collaboration. Behind them the lights of Manhattan twinkled with the squares, rectangles and circles that Felguerez' work echoed.

"Until now," Sasson began, "computer art has used the machine simply to draw geometric patterns and photographic portraits. The originality of our work is that we have fed something completely subjective into the computer."

Manuel Felguerez, who has exhibited his paintings and sculpture in Europe and the Americas for the past 20 years, has been strongly influenced by the visual images in today's technology. "The entire universe," he interpolated, "is constructed of the triangle, the circle, and the square. To create an equilibrium among such shapes is the objective of each work I create."

The interaction among these two-dimensional shapes creates spaces between them and a feeling of three-dimensionality. The result is "multiple space," which has been Felguerez' contribution to contemporary art. His achievement was crowned in 1975, when he received the Grand Prize of Honor at the Biennial of Sao Paulo, Brazil.

Sasson is a computer specialist who has participated in international conferences in London, Bucharest, Sao Paulo, Athens, Barcelona, Mexico City, and New York. Born in Colombia, he has published nearly 50 papers on such computer-related subjects as power system analysis, network theory, non-linear programming, real-time monitoring, and the application of identification theory.

How did Felguerez' search begin six years ago? "In working with geometric shapes," he said, "I began with mathematics, and in exploring the alternatives of multiple space, I had to use logic. So I began to wonder if somehow the computer, which uses logic and math, could help me. I was not sure what it would actually do.

"From the start, I intended to contact the computer through my style, for an artist's style is as characteristic as fingerprints. You can say in music, well this is Mozart, or this is Bach. And in art you cannot mistake El Greco for Rembrandt. Likewise, after more than 20 years of creating art, my work has a language, a syntax I alone use. And it seemed to me that if I would put the combination of elements that comprise my style into the computer and confine it strictly to those limits, it could be instructed to present logical alternatives to my prior works. And because it was blocked from going outside my style, there was no risk that it would influence or control me."

"I analyzed the greatest and least number of circles, rectangles, and triangles I had used in any work, and their relative size and position."

What Felguerez had discovered was a source of new ideas for his paintings and sculpture. Here was a technique for obtaining hundreds or thousands of ideas and then selecting those which seemed to have the greatest aesthetic possibilities.

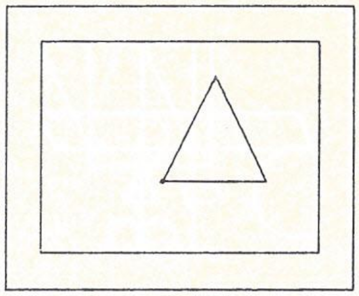
But Felguerez could not solve to his own satisfaction the problem of translating his style into a mathematical language. So in 1976, he obtained a Guggenheim Fellowship to probe the problem further.

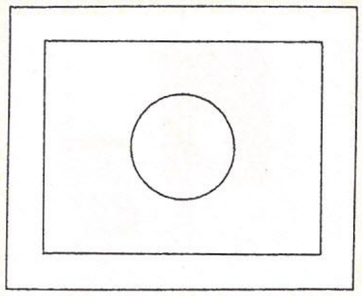
"I had reduced my geometric construction to 15 basic shapes," he continued. "I had analyzed the greatest and least number of circles, rectangles, and triangles I had used in any work, and their relative size and position. I had unearthed considerable data about my work that I knew only intuitively, but I still didn't have the answer."

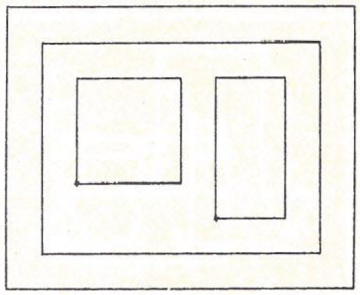
He discovered the answer one morning, when he renewed his acquaintance with Sasson during a visit to New York.

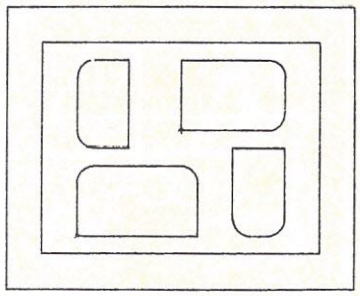
Sasson's gentle manner belies the rapier-like thrusts of his mind. Manuel says, "He set 'traps' to get me involved, but it's not really difficult to get a research man interested in a new idea."

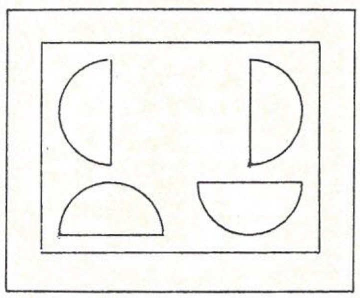
What immediately caught Sasson's eye

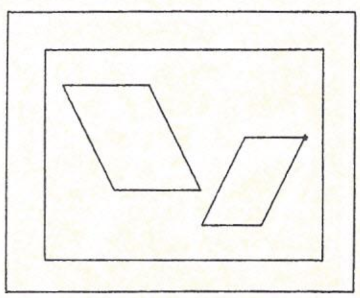


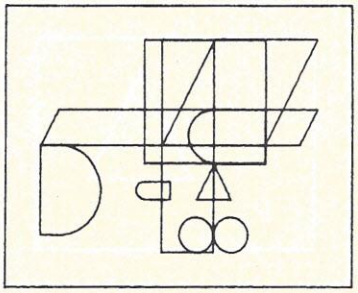




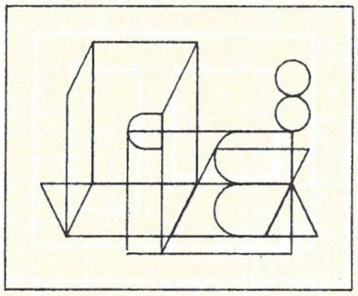


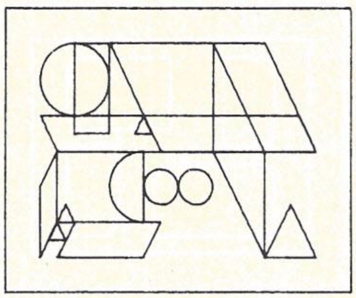


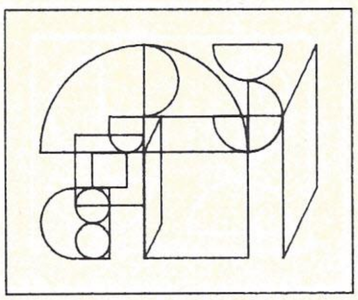


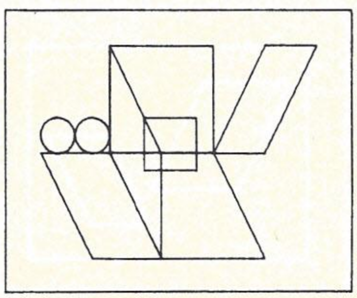


A dramatic moment came when they had summed up the 20-plus years of Felguerez' work in the first mathematical model, and instructed the computer to produce a group of new drawings.









was that the artist's problem was quite close to the problems Sasson himself had been trying to solve in his own work. "We had been concerned for a long time with analyzing how the electric power system of a utility might perform if problems arise. Most utilities in this country are interconnected, and so what's going on in the other utilities will affect what's going on in yours. But you don't have any measurements coming into your computer from those outside utilities." In the absence of facts to analyze, Sasson and his research team had turned to the system identification method and looked at what was going on at the boundaries with neighboring utilities. By collecting information at the interconnections and feeding it into the computer, his team was able to come up with a mathematical model that would explain the laws to apply to a future situation.

"I saw that Felguerez had a parallel problem, that his prior works were equivalent to how the other utilities had been operating in the past hour. Just as my prediction of what will happen – say, when a transmission line breaks down – depends on what happens outside my utility, I saw that setting up a mathematical model for Felguerez based on his past works would show him the logical direction that his own art might take in the future."

These conclusions came to Sasson during Felguerez' first visit last winter. He arose the next morning, made a few calculations, and then at breakfast asked "very timidly – fearful I might offend Manuel because I might seem to be taking something away from him – whether he would mind some suggestions." Felguerez said he would mind only if his friend did *not* offer them.

During the next three months, the pair analyzed the system that existed in the artist's mind, in the hopes of commanding the computer to produce ideas conforming to that system. Felguerez had already reviewed 300 of his works and discovered the 50 that reflected the greatest variety of shapes. Now they reduced the artist's 15 basic shapes still further to 8 – by identifying certain geometric shapes to be subtypes or mirror images of others.

Then Sasson translated Felguerez' shapes into a mathematical language. The shape itself was identified by one number, its size by numbers for the height, width, and – for three-dimensional works – depth. The position was indicated by two numbers for a square, or three numbers for a cube.

"In effect," said Felguerez, "we were summing up my artistic convictions that this circle is all right here, but that triangle has to be there – a matter of sensibility – that in a given work this square is more pleasing when larger, but that rectangle is not. We're the first to feed the computer subjective information on the sensibility of an artist."

Working by mail and telephone between Cambridge and New York, the pair reduced to numbers each of the shapes in the 50 works Felguerez had chosen. "Because each painting," said Sasson, "was described by 11 input numbers (the number and type of elementary forms) – plus 165 outputs (their size and location in space) – we needed only 11 works to create a model. That gave us a ratio of 4½ to 1 between the 50 works we had available and the 11 we needed, which provided good redundancy to identify the parameters of the model."

A dramatic moment came when they had summed up the 20-plus years of Felguerez' work in the first mathematical model, and instructed the computer to produce a group of new drawings in accord with the rules of the model. Said Sasson: "When Manuel looked at the first computer sketches and said, 'This is my world,' I knew we had achieved success."

Arbitrarily, they had limited these line drawings to 208. Felguerez then chose 54 that had artistic potential, and converted them into designs for paintings. Then these 54 new ideas were broken down into the mathematical language and fed back into the computer to enrich the model – also increasing its redundancy ratio to nearly 10 to 1. When the machine was directed to produce 208 *new* draw-

ings, Felguerez found 72 of them now merited artistic development – evidence that the model had indeed been enriched.

From the 126 computer-produced ideas, Felguerez created 18 oils and 10 sculptures for the December 1976 exhibit at Harvard University.

"In working with Manuel," Sasson explained softly, "I was not as concerned with beauty as he was. For me, the challenge was to find the process to solve the problem, and that challenge is the same, whether for a power grid or a work of art."

But the satisfaction was different. What originally attracted Sasson to the problem was the frustration of not being able to discuss the technical aspects of his work with people he met. One of the pleasant surprises of this experience, therefore, became the facility with which Felguerez grasped the problems they faced.

"In two months," Sasson went on, "Manuel understood completely what we were doing, despite all the mathematics involved. He actually pulled it out of me by asking questions. Not questions like, tell me what it's all about. He would say, well, you must be doing something like this, which means we have to do something like that. He was simply so immersed in the problem that with just a few hints he immediately saw the next logical step."

Sasson, on his part, was familiar with the art world, being married to the artist Fanny Sanin. The satisfaction he felt on seeing the exhibit at Harvard was especially rewarding.

"I felt the same way as when my first engineering paper was published. A new experience is exciting." Yet after a moment, Sasson paused, seeming to have second thoughts. In 10 years, he wondered aloud, would the emotional satisfaction revert to what an engineering paper gives him today? No, he decided, the satisfaction was different. "Maybe it's because you are accomplishing something in another field, which helps you provide a balance within yourself." It is especially rewarding, he added, when the achievement is in two areas that are so serious.

With the horizons of Sasson's life marked by music and art, as well as by engineering, and with those of Felguerez by mathematical models and space, as well as by art – we recall the men of another era. More pertinent, however, is Felguerez' conviction that art is a product of its times. He contrasted the philosophical unity shared by Michelangelo and other artists of the Renaissance with the plurality of ideas expressed by artists today. Thus, he said, "my working with a computer and geometric themes does not contradict the work of Pollock or Picasso, or even Grandma Moses."

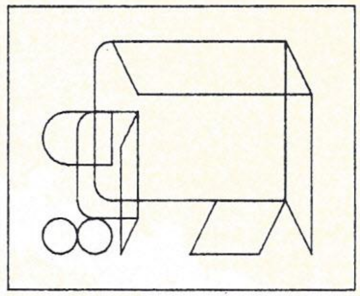
Felguerez is convinced that art is a product of its times. "Thus my working with a computer and geometric themes does not contradict the work of Pollock or Picasso, or even Grandma Moses."

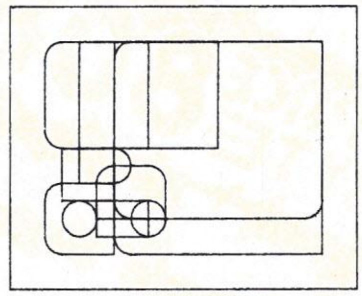
Neither Sasson nor Felguerez will predict the longevity of their art. Their only comment is that it will change as they change. Felguerez discussed the possibility of color being added to the system "by assigning a number to each color used and instructing the machine that certain colors could be used with other colors, or in contrast to other colors." Sasson described shadings and textures as another possibility.

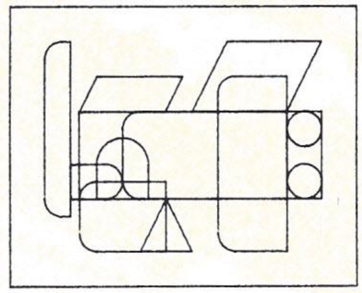
Can the computer play a role in such change if it is locked into Felguerez' current style? "In fact," the artist said, "the computer sees in the mathematical model elements that are in my work but which I myself haven't seen. So when I turn such an idea into a painting or sculpture, this is new information fed back into the computer to create a new model, which will create in turn new idea forms. Thus, we create a system, break the system, create a new system, and break that system – because only by changing does the system remain alive. Without the constant challenge there would be only repetition."

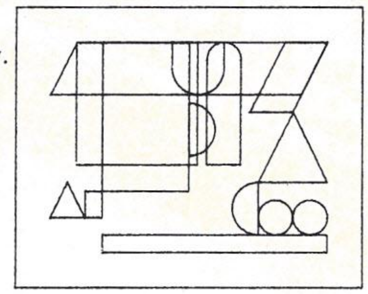
The challenge, in Felguerez' mind, is to expand art in terms of space, multiple space. For Sasson, the challenge has been to expand the science of technology in terms of man's creative sensibility.

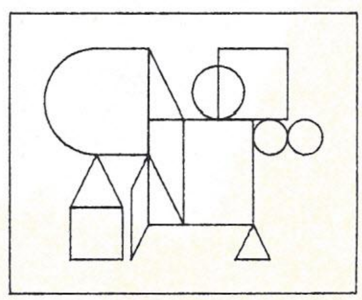
"Much has been written about cybernetics," concluded Felguerez, "and the *intellectual* capacity of the computer. But we believe that ours is the first attempt to discover what might be called its *aesthetic* capacity." He and Sasson have talked of writing a book on the subject. "We'll call it," he said, *"The Aesthetic Machine."*



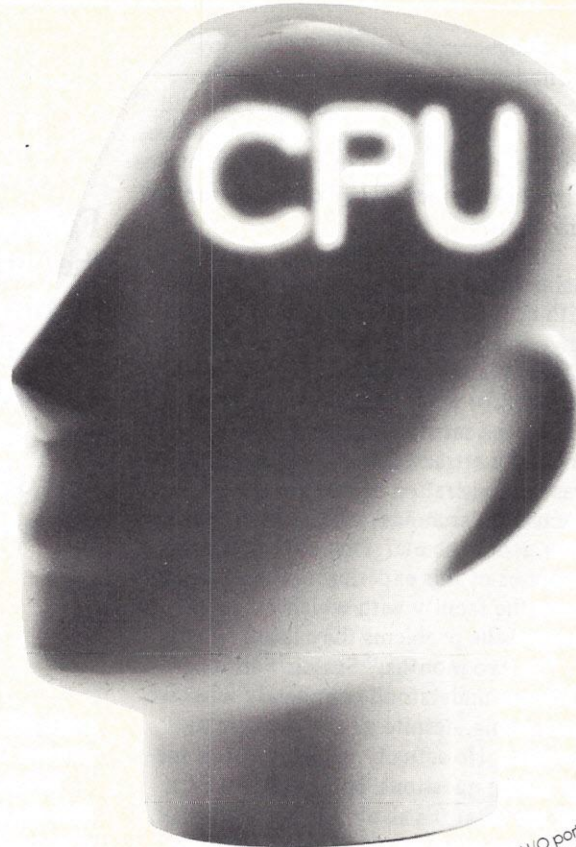




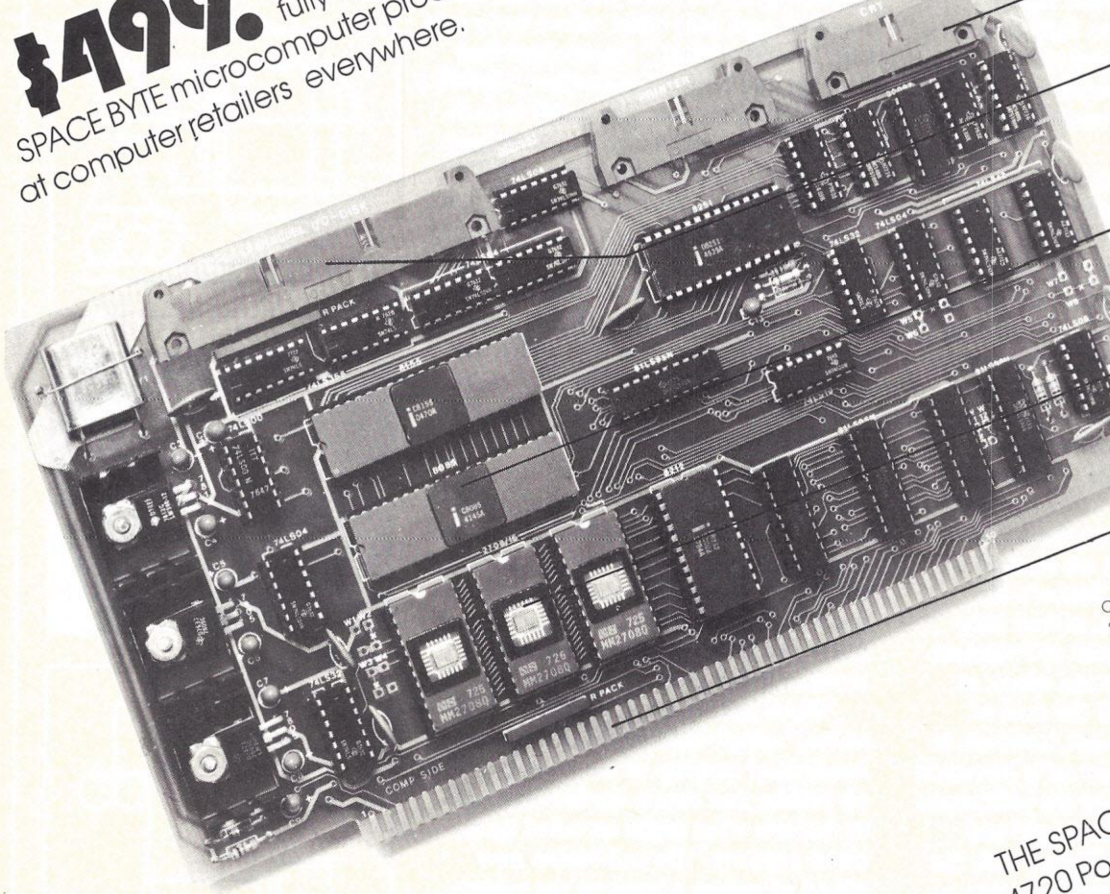




Introducing the SPACE BYTE™ 8085 CPU A self-contained computer



$499. end user quantity one.
fully assembled, burned in and tested.
SPACE BYTE microcomputer products are available now
at computer retailers everywhere.



Two RS-232C serial I/O ports—one for a CRT the other for a printer with software selectable baud rates.

Fully buffered parallel I/O ports interface directly with the ICOM® FD 3700 or Frugal Floppy disk systems

The state of the art INTEL® 8085 CPU is 50% faster than the 8080A. It operates at 3MHz, using 450 ns memory and is fully compatible with all existing 8080 software

Complete 3K system monitor with 2708* EPROMS including extensive de-bugging, FDOS Tarbell and video driver routines.
*Jumper selectable for use with 2716 EPROMS for up to 6K of on board system monitor

The all popular S100 bus...

Other features include:
256 bytes of RAM
4 vectored interrupts
programmable 14 bit binary timer/counter
optional 2708 2716 EPROM PROGRAMMER

THE SPACE BYTE CORPORATION
1720 Pontius Ave. Suite 201
Los Angeles, California 90025
(213) 468-8080

CIRCLE 62

Linear Programming: What's that?

BY PAUL WHITTINGTON

In this age of computers, many computer procedures have been developed for businesses. One such procedure, linear programming, has been used for production scheduling, routing and determining warehouse and manpower requirements. Typically, linear programming maximizes profits and minimizes costs. Let's go over some examples.

Bicycles

Suppose that you are a bicycle manufacturer and you make one model of bicycle, model A. Suppose also that the profit made on each bicycle is $5 and you can sell all that you make. How many model A's should you make? That's simple, right? Make all that you can. That's really a rather vague answer but it does lead to another question. How many *can* you make? Let's add a few more facts. It takes 2 hours to make one bicycle and only 180 hours of manufacturing time are available per month. Still pretty simple. The maximum number of model A's you can make is 180/2=90. So, in answer to the first question, you would make 90 model A bicycles each month and the monthly profits would be $5*90=$450.

Let's state this problem in a mathematically formal manner such as:

$$\begin{array}{ll} \text{MAX} & 5A \\ \text{s.t.} & 2A <= 180 \end{array}$$

This expression is read: maximize 5A such that 2A is less than or equal to 180. As we can see, this is the same as the problem stated above.

This problem is too simple to be worth using a computer to solve and is much simpler than would be found in real life, so let's complicate it a bit. Let's say that the 2 hours of manufacturing time we talked about before only referred to assembly time and that we have at most 180 hours available in the assembly department. Suppose also that it takes 2 hours to make two tires and 2 hours to make the frame, handlebars and other parts. We will also say that the time available in the tire department and frame department is 180 hours and 100 hours, respectively.

Let's write this in our formal manner as:

$$\begin{array}{ll} \text{MAX} & 5A \\ \text{s.t.} & 2A <= 180 \\ & 2A <= 180 \\ & 2A <= 100 \end{array}$$

Obviously the most restricting condition, or constraint, is the last one referring to the frame department. In order that the total time in the frame department not exceed 100 hours, only 50 bicycles can be built, because 100/2 = 50. Notice each bicycle takes 2 hours in each department. This means there will be 80 hours still available in both the tire and assembly departments. This extra time is called slack.

Two models

So far I haven't shown you anything very difficult to solve or even very useful. Well, let's back up a bit to the first problem. Let's also add that the company makes two models of bicycles, model A and model B. Model B takes 3 hours to manufacture, but the company makes $7 on each one sold. The total number of hours available still remains 180. The question now becomes, how many model A's and model B's should be manufactured:

Putting this in our mathematical form, it can be stated as:

$$\begin{array}{ll} \text{MAX} & 5A+7B \\ \text{s.t.} & 2A+3B <= 180 \end{array}$$

As a guess we might say that since the profit on model B is greater than for model A, we should make as many model B's as possible. However, since the total number of hours available is limited, what is more important than profit per bicycle is profit per bicycle per hour. In this simple example it can easily be computed as 5/2 = 2.5 for model A and 7/3 = 2.33 for model B. This shows that it would be more profitable to produce as many model A's as possible. We can check this by setting A = 0, solving for B and setting B = 0, solving for A and comparing. Doing this we can see that making only model A's results in a profit of (180/2)*5= $450, while making only model B's results in (180/3)*7= $420. So we can conclude that making as many model A's

Photograph by Gene Ritvo for Lifecycle

as possible and no model B's results in the greatest profit. (Mathematical proof that no other combinations of A and B need be considered is left to the reader).

Tires and frames

The problems are getting a bit more difficult but anyone who knows some algebra and has a little mathematical intuition could solve them without too much effort. Let's complicate things still more. Let's add that it takes 5 hours to make two tires and 1 hour to make the frame and other parts for model B. We can state our expanded problem mathematically as:

$$\begin{array}{ll} \text{MAX} & 5A+7B \\ \text{s.t.} & 2A+3B <= 180 \\ & 2A+5B <= 180 \\ & 2A+1B <= 100 \end{array}$$

Now we're getting someplace. Although it is possible to solve this problem graphically, to do so is not very straightforward and would not work if there were more than two variables. All is not lost, though. There is an algebraic method of solving this problem called the Simplex method. The method assumes a solution and repeatedly improves upon it until the best solution is reached. Without getting into specifics of how it works we will say that it will work for any problem which can be stated in our mathematical form and if the expression to be maximized (i.e., the objective function) and the constraints are all linear expressions.

According to my *Funk and Wagnalls,* a linear expression is "an expression in which all variables are in the first degree only, and which takes the form of a straight line when plotted on the Cartesian coordinate system." This simply means that the variable, called decision variables, are all raised to a power of one and each decision variable is multiplied by a constant only.

We can see that our bicycle problem falls into this category. As a matter of fact, if we were to enter this problem into our Simplex computer program, the output would look something like this:

VARIABLE	VARIABLE TYPE	VALUE	OPPR. COST
1. A	DECISION	40	0
2. B	DECISION	20	0
3. 180 – (2A+3B)	SLACK NO. 1	40	0
4. 180 – (2A+5B)	SLACK NO. 2	0	1.125
5. 100 – (2A+1B)	SLACK NO. 3	0	1.375

TOTAL PROFITS = 340

From the output we can see that we can maximize profits by making 40 model A bicycles, variable 1, 20 model B bicycles, variable 2, and the profits will be $340. All of this is very interesting and it does answer our original question, but the output tells us more than that. It also tells us that the slack variable for constraint number 1, variable 3, is 40. This means that there are 40 hours available in the assembly department that are not being used.

Possibly even more important are the entries listed under OPPR. COST (opportunity cost). Notice that the slack variable for constraint number 2, the tire department, is zero. This means that all of the available time in that department is being used. The opportunity cost for that variable is 1.125. This means that the total profits will increase by about $1.13 for every hour of additional time available in the tire department above what they are now. Similarly for the frame department, the profits will increase by about $1.38 for each increase of one hour of time available in that department.

It is obvious that an opportunity cost will be nonzero when a slack variable is zero, but this is also true for a decision variable. If, for example, it were determined that no model A bicycles were to be produced, the opportunity cost would tell us how much the profits would decrease if one model A is produced.

Roofing

Now that you're completely enthralled and totally confused, let's try another example. Take the case of a roofing contractor, Mr. Jones. The jobs he does can roughly be categorized as small, medium and large. Each job would take about a week to complete with 2 men assigned to the small jobs, 3 men assigned to the medium jobs and 4 to the large. Mr. Jones estimates that he will make about $50 on the small jobs, $70 on the medium and $100 on the large. He presently has calls to do 4 small jobs, 8 medium and 3 large. Union rules say that on each job there must be one licensed master roofer; Mr. Jones has 10 of them. Finally, Mr. Jones has 19 other employees he can send out on the jobs. Given those constraints, how many of each type of job should Mr. Jones accept in order to maximize profits?

Letting the decision variables be S, M and L to stand for small, medium and large jobs, respectively, the problem can be stated in our mathematical form as:

$$\begin{array}{llr} \text{MAX} & 50S+70M+100L & \\ \text{s.t.} & S+\ M+\ L <= 10 & (1) \\ & S+2M+3L <= 19 & (2) \\ & S <= 4 & (3) \\ & M <= 8 & (4) \\ & L <= 3 & (5) \end{array}$$

Constraint (1) takes into account that Mr. Jones has only 10 master roofers working for him, so the total number of jobs he can accept is 10. Constraint (2) is the manpower constraint for the employees who are not master roofers. Constraints (3), (4) and (5) are the job calls separated according to size.

Plugging this into our computer program, we get this output:

VARIABLE	VARIABLE TYPE	VALUE	OPPR. COST
1. S	DECISION	4	0
2. M	DECISION	3	0
3. L	DECISION	3	0
4. 10 – (S+M+L)	SLACK NO. 1	0	30
5. 19 – (S+2M+3L)	SLACK NO. 2	0	20
6. 4 – S	SLACK NO. 3	0	0
7. 8 – M	SLACK NO. 4	5	0
8. 3 – L	SLACK NO. 5	0	10

TOTAL PROFITS = 710

The output tells us Mr. Jones will maximize profit if he takes 4 small, 3 medium and 3 large jobs; the profits will be $710. Variable 7 shows 5 medium jobs will not be undertaken. Also, variables 4 and 5 show that increasing the number of employees will increase the profit. Similarly, increas-

VECTOR GRAPHIC MICROCOMPUTER SHORT FORM CATALOG

Nobody does it better.

About Vector Graphic...

Vector Graphic, Inc., since its formation in early 1976, has shown remarkable growth by adhering to principles deeply rooted in the design and manufacture of quality computers and computer products priced within the grasp of a loyal following of computer hobbyists, small businesses, schools, and research facilities.

To the exciting industry to which it belongs, Vector Graphic has continued a crusade to demechanize the computer, replacing an image of threatening machine with that of a personal tool for personal challenge, discovery and fulfillment.

As designers and manufacturers of computers, peripheral equipment and software, Vector Graphic stands among pioneers in the marketing of computer equipment offered

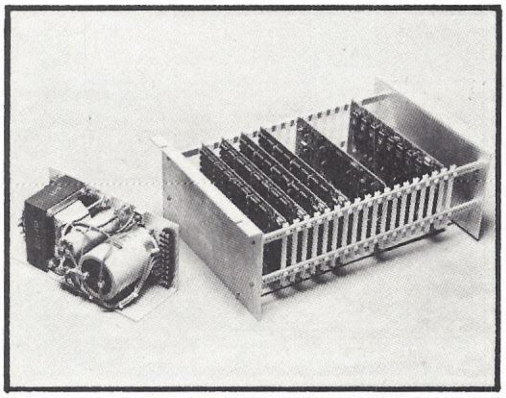
Vector 1 Microcomputer

The Vector 1 digital computer is based on the 8080A microprocessor and the common S-100 bus structure. Vector 1 has 78 basic machine instructions and a minimum instruction cycle of two microseconds. There is room for up to 64K of directly addressable memory using a parallel 8 bit word/16 bit address and up to 256 separate input and output devices can be addressed.

Rack Mount Microcomputer

A new rack mount includes a card cage, assembled and tested 18-slot Motherboard with 18 connectors, card guides and locking buttons for 18 cards. The Motherboard is fully shielded to reduce noise on the bus.

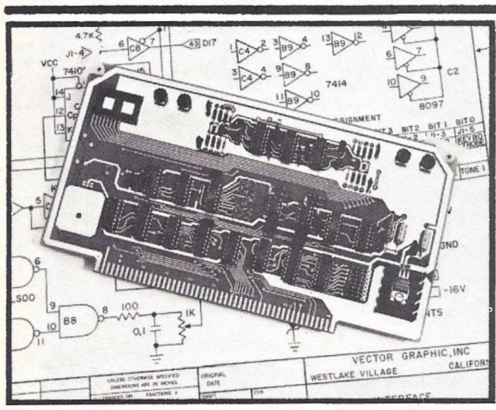
A companion power supply is available assembled or in kit form, designed for rack mounting. The 18A 8V, 2.5±16V custom supply provides power for a full 18 boards. Transformer has primary taps for 110V, 120V and 130V. Features all Class B insulation.



Dual-Stor System

The high quality Dual-Stor made for microcomputer systems, is a fully integrated floppy-disk system. It utilizes a dual floppy disk drive and controller with complete software capability in a cabinet matching the Vector 1

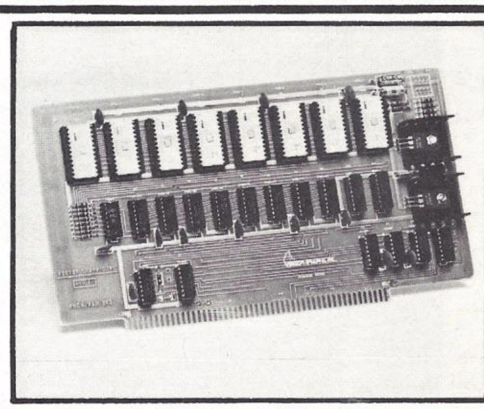
Vector Graphic's Dual-Stor disk system has a capacity of 243K bytes per disk using the standard IBM compatible recording format. Using programmed data transfer the Dual-Stor will operate with both static and dynamic memories at a standard data rate of 250K bits/sec.



Analog Interface Board

A multi-function Analog Interface Board, offered as a kit or fully assembled, permits interfacing with potentiometers, joysticks, or voltage sources. An 8-bit digital port with latch strobe can be used as a keyboard input port. Tone pulse generators also can be used to produce sounds for games or keyboard audio feedback.

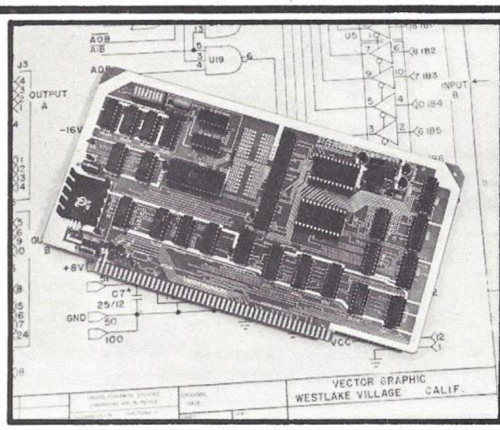
Features include four A to D inputs.



Bit Streamer I/O Board

A Bit Streamer I/O board, available assembled or in kit form, combines two parallel input and output ports, and a serial I/O port using an 8251 programmable universal synchronous/asynchronous receiver-transmitter. Communications with board circuitry is accomplished by the CPU. One parallel port also can be used as a keyboard input port.

Will operate as an RS-232 serial port using the initialization and I/O software on the Vector Graphic option C PROM.



CPU Board

Uses 8080A microprocessor with on-board real-time clock. It has 8 vectored interrupts, fully buffered data and address times. Includes 78 instructions and 2 microsecond instruction cycle time.

in low-cost kit form. From the first product — Vector 1 — to the present line comprising microcomputers, assembled and kit form boards as well as a host of accessories and software support programs, the firm's design philosophy has never varied from being dedicated to simplicity, reliability and quality.

Vector Graphic products are marketed nationally through a growing chain of more than 150 computer products stores and computer departments of recognized electronics supply outlets.

Vector Graphic's research, development and manufacturing facilities are located in Westlake Village, Calif., a few short miles west of Los Angeles, home base for many of the nation's largest computer and electronic firms.

Memorite Word Processing System

A low-cost, video-based text editing system called MEMORITE, includes the firm's Vector 1+ microcomputer with floppy disk drive, a high-resolution Hitachi 12-inch monitor, and a HyTerm printer which utilizes a Diablo 1620-3 serial print mechanism. Offers a speed of 540 words-per-minute and interchangeable printwheels.

Vector 1+ Microcomputer

The Vector 1+ is based on the 8080A microprocessor and the common S-100 bus structure, Vector 1+ has 78 basic machine language instructions and a minimum instruction cycle of two microseconds. Has up to 64K of directly addressable memory using a parallel 8 bit word/16 bit address. 256 separate input and output devices can be addressed. Vector 1+ features a front panel cutout to accommodate a Shugart SA 400 minifloppy or exact size equivalents.

Vector Graphic Microcomputer

This top-of-the-line computer offers all the features of the popular Vector 1 and Vector 1+ and extends memory capacity by accommodating two Minifloppy™ disc drives — Shugart SA 400 or exact size equivalent.

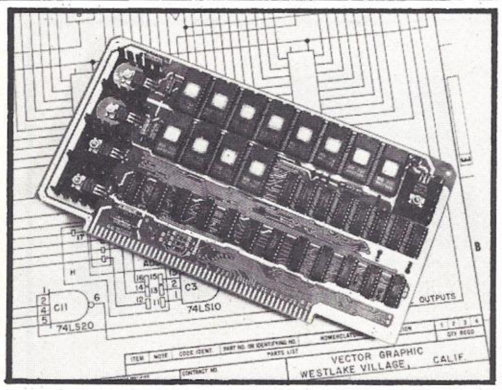
Total capacity is 65K bytes of directly addressable memory. Up to 256 separate input and output devices can be addressed.

PROM/RAM Board

PROM/RAM board, assembled or in kit form, features 1K RAM 2102LIPC low power, 450 ns access time. It has a PROM capacity for up to 2K bytes type 1702A. Stores Monitors, Bootstrap Loaders, Video Drivers.

It has stack storage; no need to relocate stack when adding memory. Uses machine language for subroutines and counters for firmware program.

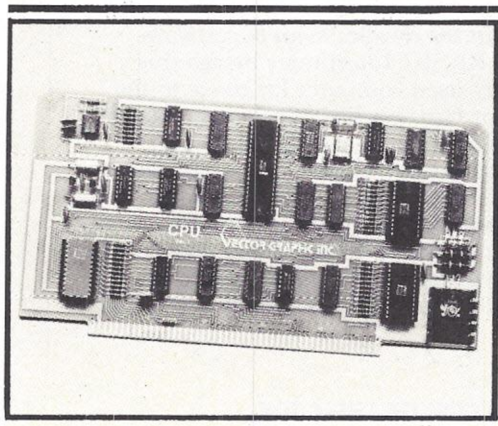
The PROM/RAM board can be used with Altair 8800 series, IMSAI 8080 and SOL as well as the Vector 1.



12K PROM/RAM Board

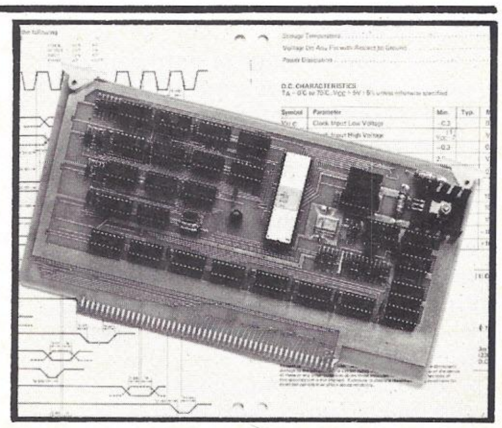
The 12K PROM/RAM Board, offered combines memory technologies of erasable programmable read-only memories (EPROM's) with high speed random access memory (RAM). 12K bytes of 2708-type EPROM's may be installed. RAM is 1K bytes of high speed static devices. Total memory is 16K bytes.

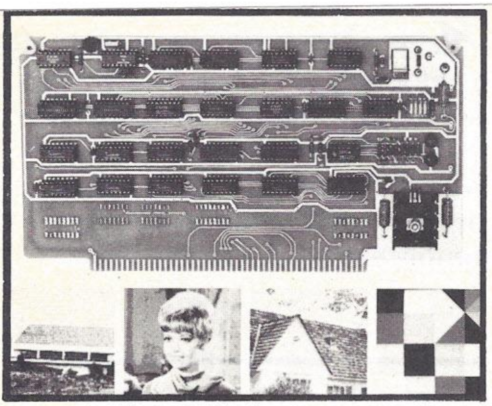
Full buffering of all inputs and outputs is provided. On-board power regulation and filtering assure stable, noise-free operation and reliability.



Z-80 CPU Board

Z-80 CPU board, offered assembled or in kit form with on-board wait state select, is jumper-selectable for operation at 2 mhz or 4 mhz. Operates standard 8080 software without modification. All Z-80 lines are fully buffered.





High Resolution Graphics Display Board

The High Resolution Graphic Display Board offers several switch-selectable video outputs. In the high resolution mode, 256H x 256V screen elements are displayed, while the gray scale mode provides 120H x 120M screen elements with 16 levels of programmable gray scale.

The board is designed to utilize a standard Vector Graphic 8K static RAM board as the refresh memory, and will interface to most rasted scan monitors.

Alphanumeric Video Display Board

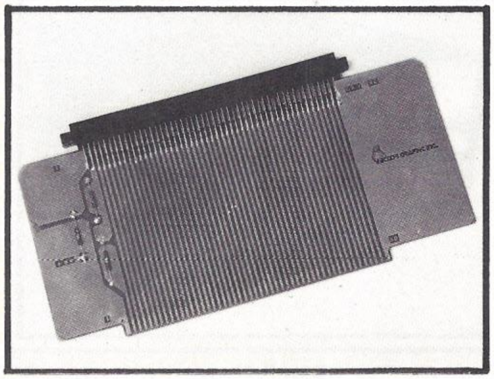
This new board will generate a video display 64 characters by 16 lines. Additional features include character-by-character reverse video, block and line graphic element generation, either may be superimposed over characters and reduced intensity field.

The board also contains a 7 x 9 character generator and independently addressable keyboard input port.

Board Extender

This highly useful accessory permits test or experimentation with Vector Graphic boards or other compatible boards without power or function disruption.

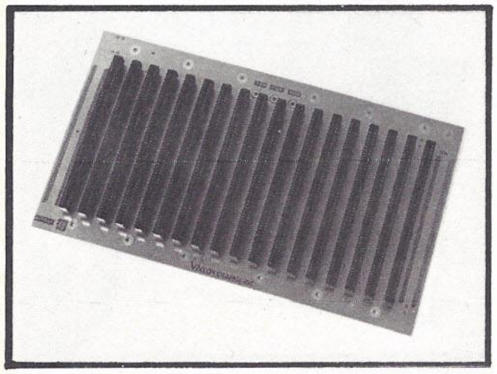
Constructed of quality G-10 material, with solder mask on both sides, the Vector Graphic Board Extender elevates the board above the chassis for easy access.



8K Static Memory Board

The 8K Static Memory Board is offered as a kit or fully tested and assembled. The board features rapid access time with no wait state. Conservative power regulators provide excellent reliability with typical power consumption of 1.5A at +5V.

Address lines buffered with Schmitt triggers; hardware 8K memory protect. Address select through top-mounted dip switch; no need to remove board.



Mother Board

Heart of the popular Vector series computers is the 18-slot, fully shielded Mother Board offered separately to the computer hobbyist and experimenter. Constructed of heavy .093 material, the board resists twisting and warping which can cause problems in board alignment.

The Mother Board also fits most rack-mount computers.

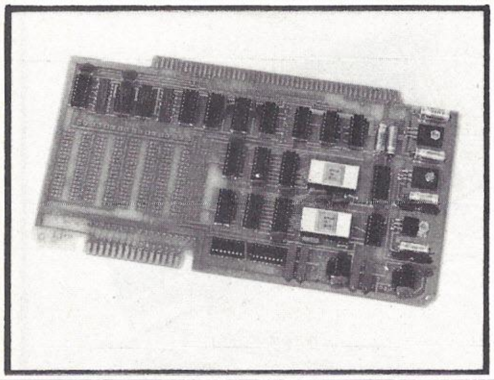
16K Static Memory Board

The 16K board is addressable as 2-8K blocks on 8K boundaries, as if two 8K memories were available. Bank selectable to allow up to one half megabyte to be addressed by the CPU. Address lines buffered with Schmitt triggers, data in and data out also fully buffered. Hardware write protection offered for each 8K block. Phantom disable feature permits use with reset-and-go PROM/RAM boards.

Precision Analog Interface Board

The precision Analog Interface Board, available as a kit or fully assembled and tested, contains 2 analog outputs and 8 analog inputs and a versatile 8-bit control port. It provides high accuracy 12-bit digital-to-analog and analog-to-digital conversion for measurement and control of analog signals.

Design uses the latest in hermetically sealed, laser-trimmed DAC's for true



12-bit (0.012%) accuracy. A variety of output voltage ranges are available and the board can operate in either monopolar or bipolar modes. A convenient patch area allows user-designed circuitry to be added.

Note: All boards S-100 bus compatible.

SOFTWARE

Vector Graphic's line of hardware now is complemented by a growing line of software support programs:

— Vector Graphic Extended System Monitor provides 25 useful commands; requires 1K bytes of PROM memory.

— Assembler/Editor program (ESP-1), offers up to 24 executive command functions for file creation, editing, debugging. Available on paper tape, cassette or diskette, ESP-1 requires only 8K of memory.

— Sourcerer, a comprehensive disassembler, can convert 8000 object code files to source code files which are much easier to understand and modify. The converted files are compatible with most 8000 assemblers including Vector Graphic's ESP-1.

— VDOS, a flexible disk operating system operating in conjunction with Vector Graphic computer systems, is an interactive operating system used in software development. VDOS obtains command from the system console, that may reference files in the VDOS directory. Files contain user-generated source language programs, object code or other user data. Offering a comprehensive set of functions, VDOS can load files by name, change name or attributes, open or close files by names and generally may be accessed by the user in the development of software.

— BASIC Interpreter brings this widely used computer language to the Vector Graphic line of microcomputers. Features include no limit or string length, string operators (concatenation, relationals, and substrings), multi-dimensional arrays, 8 commands, 22 statements and 18 functions, commands to save and load programs using the Vector Graphic Disk System.

VECTOR GRAPHIC INC.

Vector Graphic, Inc., 790 Hampshire Rd., A+B
Westlake Village, CA 91361. Tel: (805) 497-6853

ing the number of large jobs available will make the profits go up.

Wait a minute! Variable 6 has both a value and an opportunity cost of zero. What does that mean? It means that increasing the number of small jobs available will not increase the profits. Increasing the number of jobs will only increase the value of the slack variable. Such a solution is called *degenerate.*

Instead of trying to explain the significance of that now, let's change the subject slightly. So far you've taken it on faith that the opportunity cost represents an increase in profit that will occur if the resource is increased by one unit. To test that, let's increase the calls for large jobs (constraint 5) from 3 to 4 and see if the profit will increase by $10. If we do that, the output will be:

VARIABLE	VARIABLE TYPE	VALUE	OPPR. COST
1. S	DECISION	4	0
2. M	DECISION	3	0
3. L	DECISION	3	0
4. 10 – (S+M+L)	SLACK NO. 1	0	10
5. 19 – (S+2M+2L)	SLACK NO. 2	0	30
6. 4 – S	SLACK NO. 3	0	10
7. 8 – M	SLACK NO. 4	5	0
8. 3 – L	SLACK NO. 5	1	0

TOTAL PROFITS = 710

The program

Before closing, I would like to comment on the computer program itself. The program uses a lot of memory as arrays. Using the present array dimensions (20 constraints and 50 variables) and assuming the computer uses 4 bytes per number, you need 5K bytes of memory just for the arrays. Many problems will require that you increase the dimensions even further.

```
TYPE 1 IF MAXIMIZE, 2 IF MINIMIZE
?1
HOW MANY DECISION VARIABLES ARE THERE?
?3
TYPE IN OBJECTIVE FUNCTION COEFFICIENTS ONE AT A TIME
?50
?70
?100
HOW MANY <= CONSTRAINTS?
?5
HOW MANY = CONSTRAINTS?
?0
HOW MANY => CONSTRAINTS?
?0
TYPE IN <= COEFFICIENTS ONE AT A TIME
INPUT FOR CONSTRAINT NO.  1
?1
?1
?1
R.H.S. OF CONSTRAINT NO.  1
?10
INPUT FOR CONSTRAINT NO.  2
?1
?2
?3
R.H.S. OF CONSTRAINT NO.  2
?19
INPUT FOR CONSTRAINT NO.  3
?1
?0
?0
R.H.S. OF CONSTRAINT NO.  3
?4
INPUT FOR CONSTRAINT NO.  4
?0
?1
?0
R.H.S. OF CONSTRAINT NO.  4
?8
INPUT FOR CONSTRAINT NO.  5
?0
?0
?1
R.H.S. OF CONSTRAINT NO.  5
?3
VARIABLE NO.    VARIABLE TYPE  VALUE           OPPR. COST

 1              DECISION        4               0
 2              DECISION        3               0
 3              DECISION        3               0
 4              SLACK NO. 1     0               30
 5              SLACK NO. 2     0               20
 6              SLACK NO. 3     0               0
 7              SLACK NO. 4     5               0
 8              SLACK NO. 5     0               10

TOTAL PROFITS =  710

                               RANGE OF R.H.S.
                LOWER          PRESENT          UPPER

C 1              8              10               10
Z                650            710              710

C 2              19             19               23
Z                710            710              790

C 3              4              4                INF.
Z                710            710              710

C 4              3              8                INF.
Z                710            710              710

C 5              .5             3                3
Z                685            710              710
```

```
TYPE 1 IF MAXIMIZE, 2 IF MINIMIZE
?1
HOW MANY DECISION VARIABLES ARE THERE?
?2
TYPE IN OBJECTIVE FUNCTION COEFFICIENTS ONE AT A TIME
?5
?7
HOW MANY <= CONSTRAINTS?
?3
HOW MANY = CONSTRAINTS?
?0
HOW MANY => CONSTRAINTS?
?0
TYPE IN <= COEFFICIENTS ONE AT A TIME
INPUT FOR CONSTRAINT NO.  1
?2
?3
R.H.S. OF CONSTRAINT NO.  1
?180
INPUT FOR CONSTRAINT NO.  2
?2
?5
R.H.S. OF CONSTRAINT NO.  2
?180
INPUT FOR CONSTRAINT NO.  3
?2
?1
R.H.S. OF CONSTRAINT NO.  3
?100
VARIABLE NO.    VARIABLE TYPE  VALUE           OPPR. COST

 1              DECISION        40              0
 2              DECISION        20              0
 3              SLACK NO. 1     40              0
 4              SLACK NO. 2     0               1.125
 5              SLACK NO. 3     0               1.375

TOTAL PROFITS =  340

                               RANGE OF R.H.S.
                LOWER          PRESENT          UPPER

C 1              140            180              INF.
Z                340            340              340

C 2              100.           180              260
Z                250            340              430.

C 3              36             100              180
Z                252            340              450
```

Listing

```
5  REM  O ARRAY CONTAINS OBJECTIVE FUNCTION COEFFICIENTS
10  REM C ARRAY CONTAINS CONSTRAINT COEFFICIENTS
15  REM R ARRAY CONTAINS RIGHT-HAND SIDE VALUES
20  REM V ARRAY CONTAINS ORIGINAL R.H.S. VALUES
25  REM B ARRAY CONTAINS VARIABLE NO. THAT COMPRISE BASIS
30  REM X ARRAY CONTAINS -(C(J)-Z(J)) VALUES
35  REM Z ARRAY CONTAINS Z(J) VALUES
37  REM VARIABLE # IS POSITION IN OBJECTIVE FUNCTION
39  REM IF VARIABLE IS NOT IN THE OBJECTIVE FUNCTION THEN ENTER
40  REM A 0 AS THE COEFFICIENT.
42  REM FOR EACH CONSTRAINT TYPE IN VARIABLE #,COEFFICIENT
43  REM TYPE IN 0,0 WHEN ALL COEFFICIENTS FOR THAT CONSTRAINT
45  REM HAVE BEEN ENTERED.
50  DIM O[50],C[50,20],R[20],Z[50],X[50],B[20],V[20]
55  PRINT "TYPE 1 IF MAXIMIZE, 2 IF MINIMIZE"
60  INPUT M
65  PRINT "HOW MANY DECISION VARIABLES ARE THERE?"
70  INPUT A
75  PRINT "TYPE IN OBJECTIVE FUNCTION COEFFICIENTS ONE AT A TIME"
80  FOR B1=1 TO A
85  INPUT O[B1]
90  NEXT B1
95  IF M=1 THEN 115
100  FOR B1=1 TO A
105  LET O[B1]=-O[B1]
110  NEXT B1
115  PRINT "HOW MANY <= CONSTRAINTS?"
120  INPUT A1
121  PRINT "HOW MANY = CONSTRAINTS?"
122  INPUT A2
123  PRINT "HOW MANY => CONSTRAINTS?"
124  INPUT A3
125  LET K=A+A1+A2+A3+A3
126  LET A4=A1+A2+A3
127  FOR B1=1 TO K
128  FOR B2=1 TO A4
```

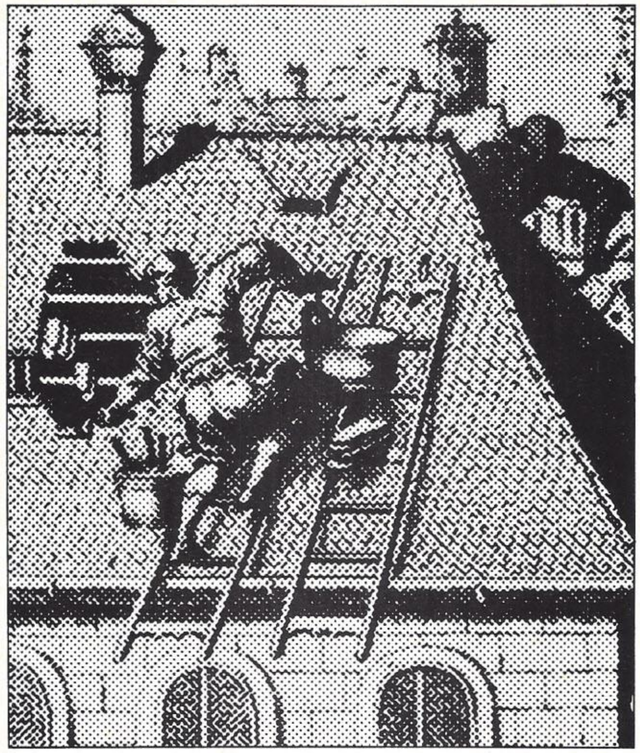
Listing - Continued

```
129 LET C[B1,B2]=0
130 NEXT B2
131 NEXT B1
132 IF A1=0 THEN 220
135 PRINT "TYPE IN <= COEFFICIENTS ONE AT A TIME"
140 FOR B2=1 TO A1
145 PRINT "INPUT FOR CONSTRAINT NO. "B2
150 INPUT B1,Y1
155 IF B1=0 THEN 165
160 LET C[B1,B2]=Y1
162 GOTO 150
165 PRINT "R.H.S. OF CONSTRAINT NO. "B2
170 INPUT R[B2]
175 NEXT B2
180 REM ***INSERT SLACK VARIABLES***
185 FOR B1=A+1 TO A+A1
190 LET C[B1,B1-A]=1
195 LET O[B1]=0
200 LET B[B1-A]=B1
205 NEXT B1
220 IF A2=0 THEN 300
223 PRINT "TYPE IN = COEFFICIENTS ONE AT A TIME"
225 FOR B2=A1+1 TO A1+A2
230 PRINT "INPUT FOR CONSTRAINT NO."B2
235 INPUT B1,Y1
237 IF B1=0 THEN 250
240 LET C[B1,B2]=Y1
245 GOTO 235
250 PRINT "R.H.S. OF CONSTRAINT NO. "B2
255 INPUT R[B2]
260 REM ***INSERT ARTIFICIAL VARIABLES***
265 FOR B1=A+A1+1 TO A+A1+A2
270 LET C[B1,B1-A]=1
275 LET B[B1-A]=B1
280 LET O[B1]=-1.E+06
285 NEXT B1
287 NEXT B2
300 IF A3=0 THEN 410
303 PRINT "TYPE IN => COEFFICIENTS ONE AT A TIME"
305 FOR B2=A1+A2+1 TO A1+A2+A3
310 PRINT "INPUT FOR CONSTRAINT NO. "B2
315 INPUT B1,Y1
320 IF B1=0 THEN 330
325 LET C[B1,B2]=Y1
327 GOTO 315
330 PRINT "R.H.S. FOR CONSTRAINT NO. "B2
335 INPUT R[B2]
340 NEXT B2
360 REM ***INSERT SURPLUS AND ARTIFICIAL VARIABLES***
370 FOR B1=A+A1+A2+1 TO A+A1+A2+A3
375 LET J=B1+A3
380 LET C[B1,B1-A]=-1
385 LET C[J,B1-A]=1
390 LET O[B1]=0
395 LET O[J]=-1.E+06
400 LET B[B1-A]=J
405 NEXT B1
407 REM ***SAVE ORIGINAL R.H.S.***
410 FOR B2=1 TO A1+A2+A3
412 LET V[B2]=R[B2]
415 NEXT B2
420 REM ***COMPUTE Z(J)***
425 FOR B1=1 TO K
430 LET Z[B1]=0
435 NEXT B1
440 FOR B2=1 TO A4
445 LET Y1=B[B2]
450 FOR B1=1 TO K
455 LET Z[B1]=O[Y1]*C[B1,B2]+Z[B1]
460 NEXT B1
465 NEXT B2
470 REM ***COMPUTE C(J)-Z(J)***
475 FOR B1=1 TO K
480 LET X[B1]=O[B1]-Z[B1]
485 NEXT B1
490 REM *** FIND PIVOT COLUMN
495 LET Y1=1
500 FOR B1=2 TO K
505 IF X[B1]<X[Y1] THEN 515
510 LET Y1=B1
515 NEXT B1
520 REM ***IF ALL C(J)-Z(J) <= 0 WE ARE DONE***
525 IF X[Y1] <= 0 THEN 680
530 REM ***OTHERWISE FIND PIVOT ROW***
535 LET Y2=0
540 LET Y3=1.E+09
545 FOR B2=1 TO A4
550 IF C[Y1,B2] <= 0 THEN 575
555 LET Y4=R[B2]/C[Y1,B2]
560 IF Y4 >= Y3 THEN 575
565 LET Y3=Y4
570 LET Y2=B2
575 NEXT B2
580 IF Y2=0 THEN 1210
585 LET B[Y2]=Y1
590 REM ***Y3= PIVOT ELEMENT***
595 LET Y3=C[Y1,Y2]
600 REM ***CHANGE PIVOT ROW***
605 FOR B1=1 TO K
610 LET C[B1,Y2]=C[B1,Y2]/Y3
615 NEXT B1
620 LET R[Y2]=R[Y2]/Y3
625 REM ***CHANGE OTHER ROWS***
630 FOR B2=1 TO A4
635 IF Y2=B2 THEN 665
640 LET Y4=C[Y1,B2]
645 FOR B1=1 TO K
650 LET C[B1,B2]=C[B1,B2]-Y4*C[B1,Y2]
655 NEXT B1
660 LET R[B2]=R[B2]-Y4*R[Y2]
665 NEXT B2
670 GOTO 425
672 IF A2=0 THEN 692
675 REM ***SEE IF SOLUTION IS INFEASIBLE***
680 FOR B1=A+A1+1 TO A+A1+A2
685 IF X[B1] <> 0 THEN 690
687 LET Y1=B1-A
688 GOTO 1195
690 NEXT B1
692 IF A3=0 THEN 715
693 IF A3=0 THEN 710
695 FOR B1=K-A3+1 TO K
700 IF X[B1] <> 0 THEN 705
702 LET Y1=B1-A-A3
703 GOTO 1195
705 NEXT B1
708 REM ***OTHERWISE PRINT FINAL VALUES***
710 PRINT
712 PRINT
715 PRINT "VARIABLE NO.","VARIABLE TYPE","VALUE","OPPR. COST"
717 PRINT
720 FOR B1=1 TO K
725 LET Z[B1]=0
730 NEXT B1
735 FOR B2=1 TO A4
740 LET Y1=B[B2]
745 LET Z[Y1]=R[B2]
750 NEXT B2
755 FOR B1=1 TO A
760 PRINT B1,"DECISION",Z[B1],-X[B1]
765 NEXT B1
770 FOR B1=A+1 TO A+A1
775 PRINT B1,"SLACK NO."B1-A,Z[B1],-X[B1]
780 NEXT B1
785 FOR B1=A+A1+A2+1 TO K-A3
790 PRINT B1,"SURPL NO."B1-A,Z[B1],-X[B1]
795 NEXT B1
800 LET Y1=0
805 FOR B1=1 TO A4
810 LET Y2=B[B1]
815 LET Y1=Y1+R[B1]*O[Y2]
820 NEXT B1
825 PRINT
830 IF M=1 THEN 845
835 PRINT "TOTAL COSTS = "-Y1
840 GOTO 855
845 PRINT "TOTAL PROFITS = "Y1
850 REM ***FIND RANGE OF R.H.S.***
855 PRINT
860 PRINT " "," ","RANGE OF R.H.S."
865 PRINT " ","LOWER","PRESENT","UPPER"
870 PRINT
875 IF A1 <> 0 THEN 885
880 IF A2=0 THEN 960
885 FOR B1=A+1 TO A+A1+A2
890 LET L=1
895 LET U=-1
900 FOR B2=1 TO A4
905 IF C[B1,B2]=0 THEN 945
910 LET Y3=-R[B2]/C[B1,B2]
915 IF C[B1,B2]>0 THEN 932
917 IF U=-1 THEN 925
920 IF Y3>U THEN 945
925 LET U=Y3
930 GOTO 945
932 IF L=1 THEN 940
935 IF Y3<L THEN 945
940 LET L=Y3
945 NEXT B2
950 GOSUB 1050
955 NEXT B1
960 IF A3=0 THEN 1220
965 FOR B1=A+A1+A2+1 TO K-A3
970 LET L=1
975 LET U=-1
980 FOR B2=1 TO A4
985 IF C[B1,B2]=0 THEN 1025
990 LET Y3=R[B2]/C[B1,B2]
995 IF C[B1,B2]>0 THEN 1012
997 IF L=1 THEN 1005
1000 IF Y3<L THEN 1025
1005 LET L=Y3
1010 GOTO 1025
1012 IF U=-1 THEN 1020
1015 IF Y3<U THEN 1025
1020 LET U=Y3
1025 NEXT B2
1030 GOSUB 1050
1035 NEXT B1
1040 STOP
1045 REM ***COMPUT RANGE OF PROFITS AND PRINT***
1050 LET Y3=0
1055 LET Y2=0
1060 FOR B2=1 TO A4
1065 B3=B[B2]
1070 IF L>0 THEN 1080
1075 LET Y3=Y3+(R[B2]+L*C[B1,B2])*O[B3]
1080 IF U<0 THEN 1090
1085 LET Y2=Y2+(R[B2]+U*C[B1,B2])*O[B3]
1090 NEXT B2
1095 LET B2=B1-A
1100 IF L>0 THEN 1150
1105 IF U<0 THEN 1130
1110 PRINT "C"B2,L+V[B2],V[B2],U+V[B2]
1115 PRINT "Z",Y3,Y1,Y2
1120 PRINT
1125 RETURN
1130 PRINT "C"B2,L+V[B2],V[B2],"INF."
1135 PRINT "Z",Y3,Y1,Y1
1140 PRINT
1145 RETURN
1150 IF U<0 THEN 1175
1155 PRINT "C"B2,"-INF.",V[B2],U+V[B2]
1160 PRINT "Z",Y1,Y1,Y2
1165 PRINT
1170 RETURN
1175 PRINT "C"B2,"-INF.",V[B2],"INF."
1180 PRINT "Z",Y1,Y1,Y1
1185 PRINT
1190 RETURN
1195 PRINT "THE SOLUTION IS INFEASIBLE."
1200 PRINT "THERE IS A DESCREPANCY IN CONSTRAINT"Y1"OF"R[Y1]"UNITS"
1205 STOP
1210 PRINT "THE SOLUTION IS UNBOUNDED."
1215 PRINT "RECHECK YOUR CONSTRAINTS."
1220 STOP
```

Wait a minute! Profits didn't increase at all. As a matter of fact, the only change has been an increase in variable 8 by one unit. Does this mean I've been lying? Not really. Remember that when variable 8 had a value of zero, it meant that all calls for large jobs had been accepted and there were none extra. That means variable 8 is one of the limiting constraints. By increasing the resource value, profits will increase. But when the value reaches a certain point, that constraint is no longer a limiting one. In fact, one of these turning points can be recognized because it is a degenerate solution mentioned earlier.

We've seen that costs are rather useless unless we know the interval within which they are valid. This brings us to the next part of the output, which looks like this:

VARIABLE		LOWER	RANGE OF R.H.S. PRESENT	UPPER
S+M+L	C 1	8	10	10
	Z	650	710	710
S+2M+2L	C 2	19	19	23
	Z	710	710	790
S	C 3	4	4	INF.
	Z	710	710	
M	C 4	3	8	INF.
	Z	710	710	
L	C 5	.5	3	3
	Z	685	710	710



C5 stands for constraint (5) and, as we can see, the upper limit is 3. So, if the number of calls for large jobs is raised above 3, the opportunity cost is no longer valid. We can see, though, that the number of small and medium jobs available, C3 and C4, can increase to infinity and it will not affect profits. It's a little easier to see why this is so with C4 than with C3. Recall that there is slack in constraint (4), which means some medium jobs are not being accepted. Logically, therefore, increasing the number of medium jobs available will not increase the number of medium jobs accepted, and consequently won't increase the profits.

If we define all the variables whose opportunity cost is zero as the Basis, the range of the right-hand side (R.H.S.) gives us the interval within which the Basis will remain the same. Note that although the variables which constitute the Basis will not change, their values will. For instance, if you increase the number of workers from 19 to 20, the variable values will be no. 1 = 3, no. 2 = 4, no. 3 = 3, no. 4 = 0, no. 5 = 0, no. 6 = 1, no. 7 = 4 and no. 8 = 0 and the profits will increase to $730.

The R.H.S. range also gives us the range in which the opportunity costs are valid. Looking back at the previous example and way back at the original output of it, we can see that increasing the number of workers, constraint (2), by 1 person, the profit does increase by $20, which is the opportunity cost.

The Z's in the output refer to the profits that correspond to the R.H.S. limits. We can see that the number of medium jobs, C4, can range from 4 to infinity with no change in profit. More importantly, in this case, we can see from C2 that if Mr. Jones can hire 4 more workers, all else remaining the same, his profit changes from $710 to $790 – an increase of $80 (which is 4x opportunity cost).

Up until now we've been talking only of increasing profits. Often we're more concerned about reducing costs. You can reduce costs in two ways. If all the variables in the objective function are costs, you can tell the computer that the problem is one of minimization and all the details will be taken care of for you. Another method, probably more useful, is to recognize that you can minimize a function by maximizing its negative. This means you can treat costs as negative profits and maximize. No matter which method you use, the final answer will be the same, except that the total cost will equal the negative of the total profits.

The only constraints we've dealt with are <= constraints. As you may have guessed, there are also => and = constraints. A => constraint would occur if there is a lower limit on a resource (such as having to manufacture at least 7 model A bicycles per month). Associated with these constraints is a surplus variable which is similar to the slack variable. It tells to what extent the minimum requirement has been exceeded. Also, the = constraint will occur when a requirement must be met exactly (such as in a manufacturing process whose useable output is exactly some percentage of the input).

Besides degenerate solutions, two other types of solutions need to be discussed. The first is the *alternate optimal solution,* which occurs when more than one solution will result in the same maximum profit. You can tell when it occurs, since one or more of the variables will have both a value and an opportunity cost of zero. Wait a minute! That's the same as for a degenerate solution. How do you tell the difference? Well, if the number of variables whose values are not zero equals the number of constraints, you have an alternate optimal solution. If the number of non-zero variables is less than the number of constraints, the solution is degenerate. Unfortunately, the computer program will not

tell what the alternate solution is, but at least you can be aware that one exists.

The last special type of solution is called an *infeasible solution.* This occurs when there is no solution which will satisfy all of the constraints. The computer will tell you explicitly when it occurs, which constraints are violated and by how much. It normally occurs only when the problem has been formulated incorrectly.

Bicycle kits

As the final example let's modify our original bicycle problem. Let's say that the one model that you make is sold either fully assembled or as an unassembled kit. The factory is still separated into three departments (A, B and C), which are the tire, frame and assembly departments. You have two workers, Joe and Sam. This table shows how many units they can process in each department per hour and their hourly wages:

	Tires	Frames	Assembly	Wages
Joe	10 per hour	10 per hour	7 per hour	$4.75 per hour
Sam	12 per hour	14 per hour	7 per hour	$5.00 per hour

The anticipated demand is 500 assembled bicycles and 250 kits for the first month, 1000 assembled bicycles and 250 kits for the second month, and 1500 assembled bicycles and 500 kits for the third month. You have 100 sets of tires and 75 sets of frames in inventory now. The monthly inventory costs are $1.00 per set of tires and $1.50 per set of frames. There are no assembled bicycles kept in inventory and inventory costs are paid at the end of each month.

The sales price, less fixed and raw material costs, is $10 for each assembled bicycle and $7 for each kit. You'd like to find how many hours each man should work in each department during the three months and how many parts will have to be inventoried at the end of each month. Assuming there are 22, 22 and 21 working days in each of the three months, we can represent the problem as:

$$
\begin{array}{ll}
\text{MAX} & 31787.5Z-4.75A_{11}-4.75B_{11}-4.75C_{11}-4.75A_{12} \\
& -4.75B_{12}-4.75C_{12}-4.75A_{13}-4.75B_{13}-4.75C_{13} \\
& -5A_{21}-5B_{21}-5C_{21}-5A_{22}-5B_{22}-5C_{22}-5A_{23}-5B_{23} \\
& -5C_{23}-1I_{A1}-1I_{A2}-1.5I_{B1}-1.5I_{B2}
\end{array}
$$

s.t.

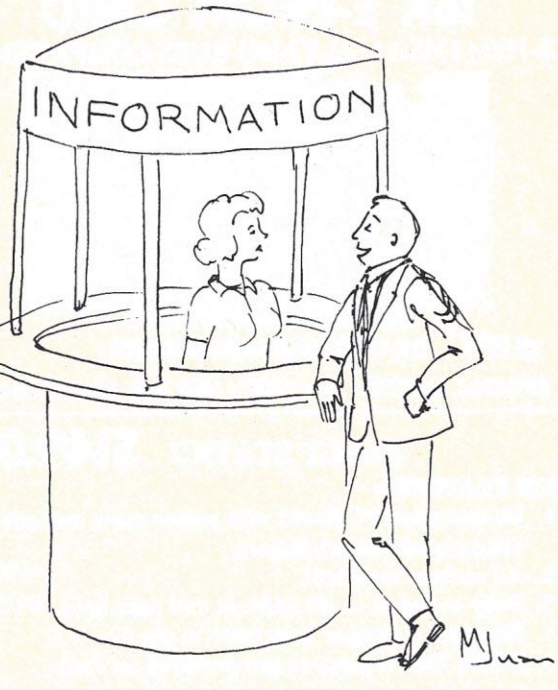
$$
\begin{array}{rlll}
A_{11}+\ \ B_{11}+ & C_{11} & <= 176 & (1) \\
A_{12}+\ \ B_{12}+ & C_{12} & <= 176 & (2) \\
A_{13}+\ \ B_{13}+ & C_{13} & <= 168 & (3) \\
A_{21}+\ \ B_{21}+ & C_{21} & <= 176 & (4) \\
A_{22}+\ \ B_{22}+ & C_{22} & <= 176 & (5) \\
A_{23}+\ \ B_{23}+ & C_{23} & <= 168 & (6) \\
10A_{11}+12\ A_{21}- & I_{A1} & = 650 & (7) \\
I_{A1}+10A_{12}+12\ A_{22}- & I_{A2} & = 1250 & (8) \\
10B_{11}+14\ B_{21}- & I_{B1} & = 625 & (9) \\
I_{B1}+10B_{12}+14\ B_{22}- & I_{B2} & = 1250 & (10) \\
7C_{11}+\ 7\ C_{21} & & = 500 & (11) \\
7C_{12}+\ 7\ C_{22} & & = 1000 & (12) \\
7C_{13}+\ 7\ C_{23} & & = 1500 & (13) \\
 & Z & = 1 & (14) \\
I_{A2}+10A_{13}+12\ A_{23} & & => 2000 & (15) \\
I_{B2}+10B_{13}+14\ B_{23} & & => 2000 & (16)
\end{array}
$$

The variables A, B and C stand for how many hours each man works in each of the departments. The subscripts designate which man and which month we are referring to. So A_{11} stands for how many hours the first man, Joe, worked in department A in the first month. Similarly, I_{A1} stands for how many units from department A (sets of tires) will be inventoried at the end of month 1.

The objective function is separated into three parts: revenue, wages, and inventory costs. The first term is a combination of the total expected revenues less the first month's inventory costs (i.e., 7* total kits + 10* total assembled -1*100 -1.5*75) times a dummy variable. The computer program doesn't let us have constants in the objective function or on the left-hand side of the constraints. We overcome this by creating a dummy constraint (14) and multiplying our constant times the dummy variable, Z.

Constraints (1) through (6) are how many hours each man works per month. Constraint (7) refers to how many sets of tires are made the first month. The left side, the number of sets manufactured less the number inventoried, must equal the demand for assembled and kit bicycles less the number already in inventory. Similar logic gives the other constraints. You can expand the problem to include more workers, departments, products, and months, and modify it to handle overtime and union rules.

Linear programming cannot solve many problems involving economies of scale or in which the variables are not all linearly related. But linear programming is often a good approximation that provides a starting point for many types of decision making. ▮



"What's 47,806 in hexadecimal?"

A/D Conversion

by STEVE POLLINI

The physical world does not provide information that the computer can digest by itself. Temperature, light levels and human speech all take values on a continuum, but the computer can process only discrete, binary numbers. To turn the continuous, analog signals into binary, digital signals, you need an analog-to-digital converter.

On last night's late report the weatherman said that the temperature would reach 101°F today, a record high for the month. A neighboring town is supposed to experience a high of only 95°F. Have you noticed that the temperature report or forecast is given in *discrete* numbers? When was the last time you heard the weatherman say: " . . . tomorrow there will be a slow, steady rise from 75.0031°F to 98.1087°F with a rate of change of 4.25 degrees per hour . . .", etcetera, ad nauseum?

Except when some unusual change is expected, like the passage of a weather front through an area causing a temperature drop when a rise would be normal, the temperature is seldom reported as a slow, continuous analog change over time. The temperature can rise very slowly; a graph of the action would look like Figure 1.

FIG 1

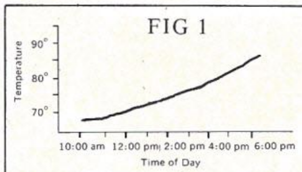
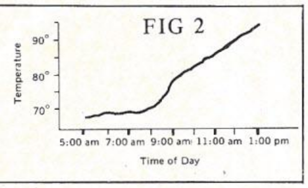


FIG 2



However, the temperature might rise slowly for a while, as at dawn, then rise very rapidly, as when the sun rises or comes out from behind clouds. A graph showing this sort of change might look like Figure 2.

FIG 3

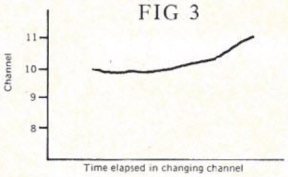
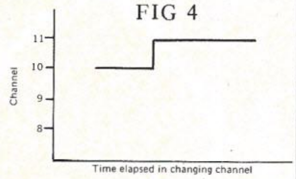


FIG 4



It isn't often that you need to know this much detail about the situation.

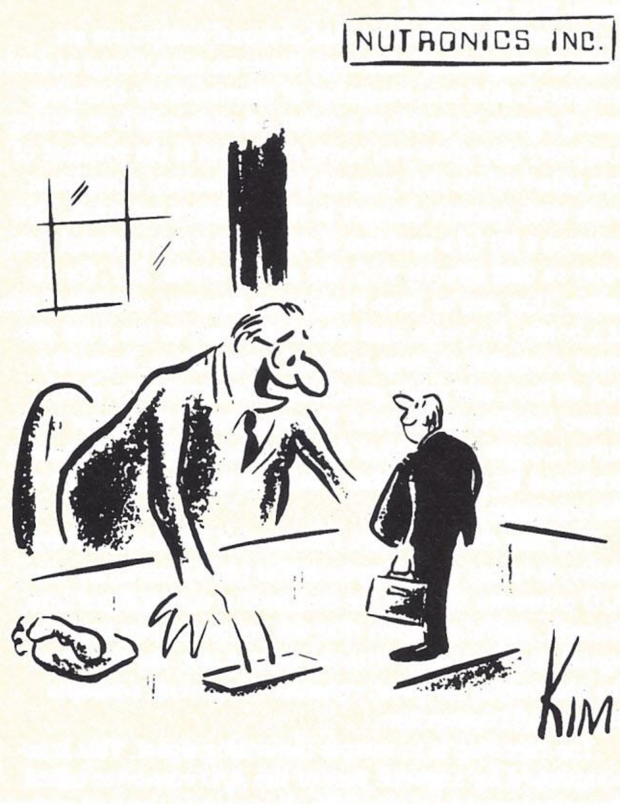
The important point is that temperature changes continuously, slowly, and not in sudden, discrete steps. It does not jump from 75° one second to 80° the next, and to 65° the next. We refer to information describing the continuously changing phenomena as "analog" information. By contrast, "digital" information is the sort of thing you get when you change channels on your television set. You do not start at channel 10 and slowly, in infinitesimally small increments, approach channel 11. A graph showing a channel change from 10 to 11 looks like Figure 4, not Figure 3.

At one moment, you are on channel 10, and at the next, you are on channel 11. Snap! (Purists will insist, rightly, that even the channel change cannot take place in no time *at all,* so the event is really analog with a rise time – that vertical jump – so steep that it is difficult to detect. Still, for practical purposes, we can, and do, treat the change as a real, digital step function.)

You may have noticed that personal computers are "digital computers," although there are such things as analog computers that sum a number of continuously changing analog signals to produce a value. (A slide rule is an analog computer that gives you some value that is the product of two other values that you set.) Most personal computers work with discrete changes in value, binary numbers. This presents a bit of a problem if you want your personal computer to deal directly with the analog world. Say, for example, that you've been keeping an amateur weather station as a hobby for years, and would like now to use your personal computer to keep track of temperature, humidity and all that, and make predictions automatically about future weather conditions. The problem is that your computer accepts only digital data and the analog information produced by your weather station is incomprehensible to the computer.

The way to make them friends, and let them converse easily, is to use an analog-to-digital (A/D) converter. An A/D converter changes the continuous analog signal into discrete, digital numbers that the computer can easily recognize and deal with. Without getting into the details of the electronics involved, consider the general concepts behind A/D conversion.

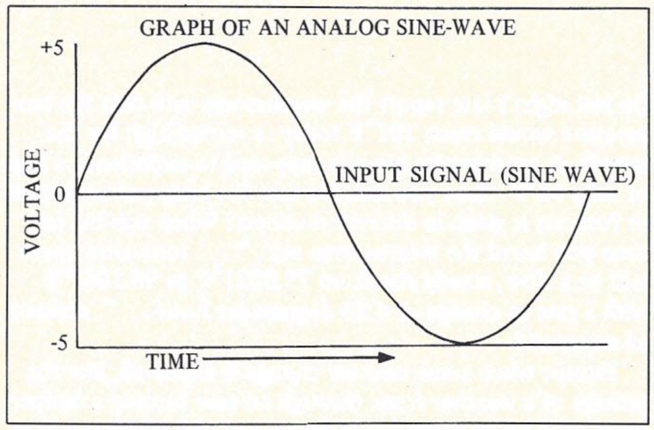
The first step is to *sample* the analog signal. This is accomplished by a special part of the A/D converter called the *sample-and-hold amplifier.* It measures the analog signal at a point in time, then holds this sample measurement steady while the rest of the system converts it to a digital number. Once that single conversion is made, the A/D sig-

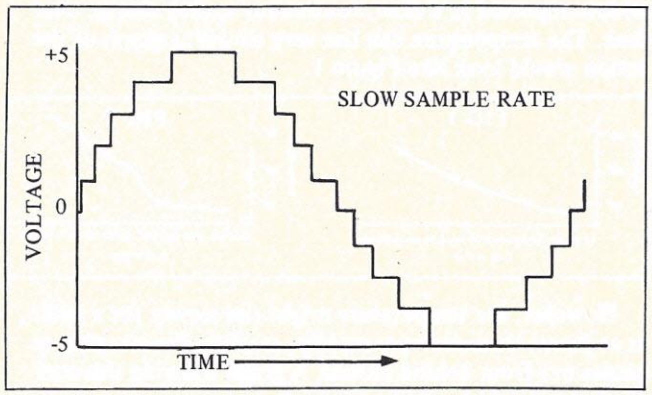


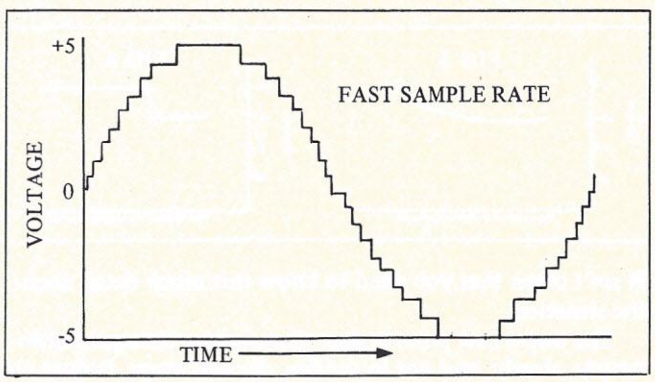
"So – you're the new micro-computer expert."

nals the computer that a digital number is available. The computer then takes the discrete binary number and stores it in memory for possible further attention. The A/D meanwhile turns back to the analog signal and grabs another sample. The frequency of sampling may be as high as millions of times a second. Usually, the sampling rate is determined by a clock, though in some applications the A/D may take a sample when instructed to by the computer, which knows what it needs and when it needs it. In any case, the process is basically the same: the sample-and-hold amplifier measures the analog signal and holds the value for conversion to a discrete number. Then the A/D tells the computer a number is available, hands it over, and turns back to the analog signal for more.

The lower graphs represent what the A/D converter detects when it examines the analog signal over a period of time. (In this case, the time is one second, and since one full cycle of the sine-wave is represented, the A/D is looking,







conveniently, at a 1 Hz signal.) The computer receives from the A/D a string of numbers, each indicating the height of a discrete stair-step. The numbers begin with zero, where the graph begins. The numbers get larger gradually, peaking at +5; then they get smaller, dropping through zero all the way to -5. From there, they rise to zero again.

If you count steps, you'll see that there are only 25 conversions for one second of analog signal input. Since the analog signal is at a very low frequency (at least in electronic terms; if the signal represented the swing of the pendulum of a big clock, the rate wouldn't seem slow), twenty-five samples a second gives a good enough digital representation for a computer to deal effectively with the sine-wave. If the original signal were at ten thousand cycles per second, the 25 samples-per-second would not give the computer a useful description of that signal. On the other hand, if the sampling rate were 10,000 times per second and the analog signal being examined were this same 1 Hz sine-wave, the stair-steps would be so small that the digital representation of the analog signal would be extremely smooth.

The sampling rate selected depends on a number of factors, such as nature of the original analog signal and the rate at which the busy computer can accept data. The art of matching A/D systems to computers is a fine one, and cost is not the least important consideration. Analog-to-digital (and digital-to-analog, D/A) equipment tends to rise in cost with the speed at which it operates, among other factors.

Your weather station? By electronic standards, weather changes slowly. The cost and complexity of your prediction system will depend more on the number of factors you wish to consider than on necessary conversion and processing speeds.

A/D conversion is growing more common as computer processing grows more common. Television systems ordinarily operate entirely with analog signals, converting light levels to changing electrical levels, then back to light again on the face of the television set. However, television signals are increasingly being processed digitally, and the pictures you see may be represented somewhere along the line as a string of discrete numbers.

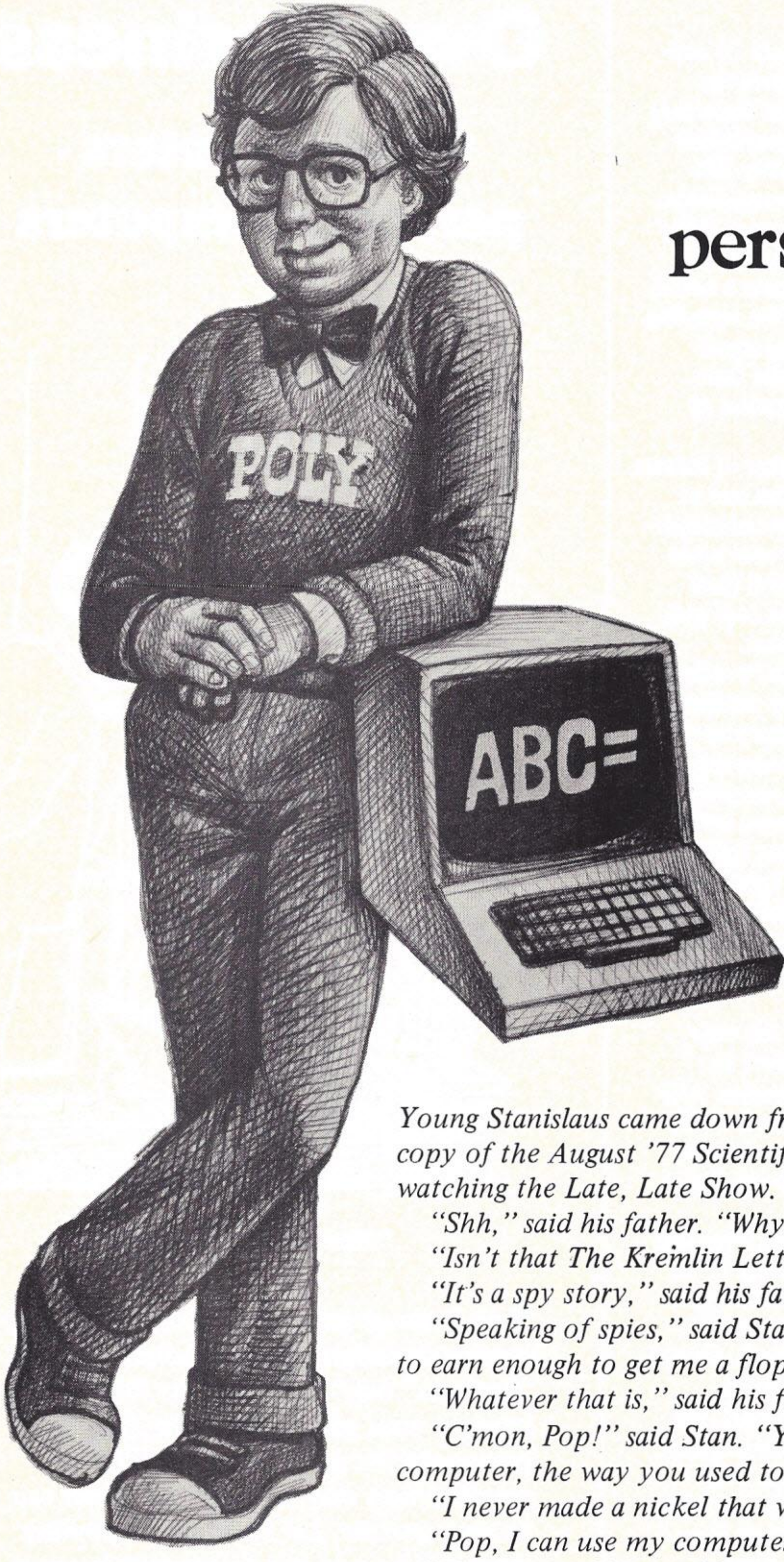
And sound? You may be looking forward to the time when you speak aloud to your computer to control its action and to give it information on which to operate. A/D conversion is clearly a necessary part of that process in the first speech-recognition systems that are reaching the market now, and will presumably remain part of the process as more sophisticated equipment is developed.

However, note that mere conversion of a signal from one form to another does nothing to reveal the *meaning* of the signal. Even if your computer can get the signal when you speak to it, it may not get your message. If you cure it's deafness, it may still be unable to understand Greek, no matter how clearly and distinctly you speak the language.

It is possible, without any scientific breakthroughs, for your own computer to communicate with the real world of analog signals in a thousand practical applications. The trick is to avoid saturating the poor computer with more information than it can use. You probably don't want to explain that the temperature is climbing gradually from 74.0968732° to 75.4975853°. Your computer will like it better if you say: "The temperature is 74.1° . . . the temperature is 75.5° . . ." or binary numbers to that effect. pc

CLOAK AND DAGGER
with your personal computer

BY TIMOTHY PURINTON



With the mails getting slower and more expensive and electronic communication getting faster and cheaper, business and personal correspondence moves inexorably towards the electronic transmission facilities.

One inhibition to the enjoyment of wires and wireless is the ease of tapping them. Private information is much too susceptible to becoming public. The solution? Put the transmitted information into a cipher to which the proper recipient alone has the key.

Ciphering has been popular throughout recorded history. Priests have protected their mystique from the curious layman; military men have hidden their intentions from the enemy; businessmen have end-run their competitors.

As microcomputer operators – if your system will generate pseudo random numbers from a seed – you have

Young Stanislaus came down from his room carrying a dog-eared copy copy of the August '77 Scientific American, and found his father watching the Late, Late Show. "What's on?" said Stan.

"Shh," said his father. "Why aren't you in bed asleep?"

"Isn't that The Kremlin Letter? Terrible picture."

"It's a spy story," said his father. "Can't be all bad."

"Speaking of spies," said Stan, opening the magazine, "I'm trying to earn enough to get me a floppy."

"Whatever that is," said his father. "Shh."

"C'mon, Pop!" said Stan. "You want me to earn money with my computer, the way you used to with your lemonade stand. . ."

"I never made a nickel that way. That was your grandfather."

"Pop, I can use my computer to encipher secret messages. I can sell the service, like a public stenographer. I can set up a network with guys in other places. We can be the International Cloak & Dagger Service Bureau. . ."

His father walked across the room and snapped off the TV set. "What?" he said. . .

Illustrations by Kim Behm

at your command a way to put messages into unbreakable cipher automatically.

Over the centuries, many an ingenious ciphering method has been developed, each thought by its inventor to be unbreakable, each expected to produce a fortune.

But the truly unbreakable cipher is rare if it exists at all. The only person known to have reaped a fortune at it is Boris Hagelin, whose portable ciphering machine was ready for mass production at the outset of World War II.

All ciphers are built on two principles: transposition, and substitution. The cipher hides the meaning of a communication through a systematic disorder that the intended recipient can set straight. Extreme complexity is not counted a virtue, because the person primarily inconvenienced by complication is the one to whom the message is addressed.

The simplest transposition is exemplified in a schoolboy's effort – "YRAM, I EVOL UOY. YLLIB." A bit more sophisticated would be a transposition achieved by writing across the turns of a strap wrapped around a stick, and deciphered on a stick of the same diameter – a popular primitive system. A simple substitution is the Caesar Shift, used by Julius himself:

plaintext: abcde. . . wxyz
CIPHER: DEFGH . . . ZABC

Caesar went from plain into CIPHER by shifting the alphabet three steps. His brief report of success in Gaul – *veni, vidi, vici,* as every schoolboy used to know – became YHQL, YLGL, YLFL. His correspondent in Rome consulted his own alphabetic slipstick and brought the message back to plain Latin.

Elementary as it is, this kind of substitution shift leads to the unbreakable microcomputer enciphering method.

Julius made his cipher too easy to solve. He shifted every letter exactly three steps, year after year – apparently flaunting his disrespect for the analytical powers of the enemy, or perhaps because he was confident they weren't too hot in Latin anyway. With little work, the Wrong Eyes could discover the three-step key, and since the key to one message was the key to them all, Caesar might just as well have written in the clear.

The standard approach for the cryptographer faced with a substitution cipher is to tally the frequencies of the characters used in the message, and to compare it with the table of frequencies of the characters used in plaintext in the same language. In English, the letters used most heavily are those that make up the linotypist's gibberish space-filler, *etaoin shrdlu* – from the two left-hand columns of his keyboard. Any cipher, such as the Caesar Shift, in which the same character stands for another throughout the message (or the user's lifetime, in Caesar's case) can be solved easily by frequency analysis.

We make the giant leap from the childishly-obvious three-step shift to the unbreakable all-time champion by introducing *random numbers,* which give us what the cryptographer calls a "One-Time Pad."

Suppose encipherment starts with the alphabetic slipstick positioned as shown above, with CIPHER D at plaintext "a." Let's encipher the plaintext word "act." The first letter of the plaintext message, "act" would become D. Now for the next letter, *shift again,* "randomly," to create systematic disorder of greater subtlety. Say that for the second letter, CIPHER O is indexed at "a" and our plaintext "c" comes out CIPHER Q. Now shift again, indexing C at "a," "t" becomes R; plaintext "act"=DQR. The key we've used is DOC.

This is an unbreakable cipher. Even a long message coming before the Wrong Eyes can provide no basis for frequency analysis, nor for study of common diphthongs, nor for any other cipher-cracking method. And even if some part of the message were to become known ("We know it concerns *shipments, prices, availabilities* – look for these. . ."), still no *key* could be revealed that could unscramble the meaning. That calls for rational relationships between plaintext and the characters substituted. But our shifts are *random.*

The One-Time Pad is inconvenient. The key is necessarily always as long as the message itself. And the key must be safely conveyed somehow from the sender to the intended recipient. Still, inconvenient or not, the One-Time Pad approach is constantly used today for special messages between high military commanders, and between governments and their agents. The "pad" is commonly a book of pages of random numbers. When you've used a page once, tear it out and destroy it. The recipient deciphers with the corresponding page in his copy of the book, and does the same. No repetitions. And no way to crack the cipher if the pads themselves are kept secure.

Enter Stanislaus and his microcomputer (the Compal-80 with Extended BASIC). Suppose Stan roughs-out a program for that language:

INPUT a letter of plaintext. Call it A$.

A certain Histiaeus, member of the Persian court some 2500 years ago, planned a palace coup. Needing outside support, he sought to communicate secretly with his son-in-law Aristagorus, the "tyrant of Miletus," in neighboring Greece. ("Need help quick. Am seizing Persia. How's the family? H.")

So Histiaeus "summoned his most trusted slave," the historian Herodotus tells us, "shaved his head, and tattooed the message on his scalp." ("Hold still, Rastus. . . There now. You just sit in the closet till your hair grows back.")

In due course, the slave slipped across into Greece. ("If a Persian stops you, you know nothing. He's a Greek, a big Greek guy. You can't miss him.") To Aristagorus, he spoke the key as he had been taught it: "Shave my head and look thereon."

The coup was a great success, but this system of secret writing never did catch on.

Find the numerical value of the letter A$ – its place in the alphabetical hierarchy. ("a" is 1, "b" is 2, etc. Go on into the numerics, if you like: Ø is 27. . . 9 is 36.) Let the numerical value of A$ be called X.

Generate a random number and add it to X. Call the total K.

Look again at the alphabetical hierarchy, and find what's at position K. Call the character there B$.

PRINT B$. It's the first letter of the CIPHERTEXT.

Go back and INPUT the next letter of plaintext, and continue.

That does the enciphering.

To decipher, Stan's correspondent goes through the same steps, but finds K by *subtracting* the same random numbers.

Here's the program, as shaped up by young Stansilaus:

```
99 CLEAR 100
100 B$="ABCDEFGHIJKLMNOPQ
    RSTUVWXYZ0123456789"
120 READ A$
130 FOR X=1 TO 36
140 IF MID$(B$,X,1)=A$
    THEN 160
150 NEXT X
160 R=INT(36*RND(1))+1
170 K=X+R
171 IF K<37 THEN 180
172 K=K-36
180 PRINT MID$(B$,K,1);
200 GOTO 120
300 DATA M,E,S,S,A,G,E,X,
    G,O,E,S,X,H,E,R,E
```

Not bad, eh, for a 10-year-old hockey player?

Stan explains that he needed 99 CLEAR 100 because his long B$ on the next line seemed to be clobbering the run. Actually, the B$ was originally twice as long, because Stan repeated it so that in the decipher mode his K would be able to find a character even with a value over 36. But the adjustment at 171-172 brought everything back into range, and B$ could be cut to the single 36-character string.

Lines 130-150 mean "find the character in B$ which agrees with the A$ character read from the DATA. Its position in the B$ hierarchy will be the value of X. When that is determined, proceed to line 160."

Line 160 calls up a random number between 1 and 36, inclusive. Line 170 adds X and the random number to produce K, the value of the desired CIPHERTEXT substitute. Line 180 finds that character in the lineup and prints it. Line 200 hustles back to go to work on the next letter.

In 15th century Germany, Johannes Trithemius – who had changed his name from Heidenberg, and had declared that the world was created in 5206 B.C. (Before Carbon-Dating) – announced a low-cost communication system with a high degree of security: "To convey a message to a desired recipient within twenty-four hours, say it over the image of a planetary angel at a moment determined by (complicated astrological) calculations; wrap the image up with the image of the recipient; bury them under a threshold; say the proper incantations, ending with 'In nomine patris et filii et spiritus sancti. Amen."

The decipher mode is exactly the same, except for the switching of + and -, and > 0 for < 37, at 170-172:

```
170 K=K-R
171 IF K>0 THEN 180
172 K=K+36
```

But how does the decipherer get the same random numbers as the encoder? The Compal's BASIC, in response to RND(1), will deliver a value somewhere from .000001 to .999999. It comes from an algorithm built into the BASIC program itself, and the numbers are not truly *random*: they come out as the same sequence every time.

Omitted from the listing above is a pair of lines Compal needs, to harness the random numbers:

50 INPUT "WHEN"; ZZ

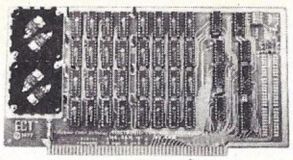
60 XX=RND(-ZZ)

This INPUT ZZ is the "seed" – and the key to the cryptogram. It gives the algorithm a *starting place* – and the same sequence of numbers will be generated in response to the same seed. It provides *plenty* of numbers to handle long messages without a telltale repetition. (Stan experimented, built a program to search for a repeat of the opening number in a sequence. Though the machine didn't tire, Stan did; he turned it off at 80,000.) The word at Compal is that the seed stretches the algorithm's yield to something between four and five million. Stan says he figures the enemy could work all afternoon guessing at seeds and never hit the one that would start the cycle, in the right place among four or five million, to make his messages come out in the clear.

But Stan expects to have no trouble communicating his compact key/seed to his correspondent. They might agree, for example, that the key for a message sent on Monday would be the closing price of Du Pont Pref A as of the previous trading day. With a good financial page available to each, they'd both know the key is 63.75.

Stan's attention to this project is inspired by Martin Gardner's discussion in *Scientific American* of a ciphering method worked out by a group at Massachusetts Institute of Technology and their computer. It uses a huge prime number raised to its 9007th power, plus an algorithm, and it produces a message that looks like this:
060918192000191512220518002309
1419001514050008211404180504
0004151212011819

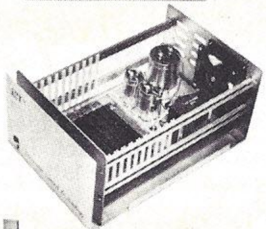
That's just the teaser: you might be able to see that it translates to FIRST SOLVER WINS ONE HUNDRED DOLLARS. But the message the article asks us to solve is longer and broken into four-digit groups. There may still be time for you to get into that competition. With the fastest computer available today and the best known algorithm, to ring the changes that will crack that cipher take an estimated 40 quadrillion years. Unless this issue of *Personal Computing* is later than usual, someone's still working on it.



16K RAM

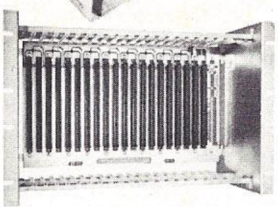
FULLY STATIC

$450 INTRODUCTORY KIT PRICE



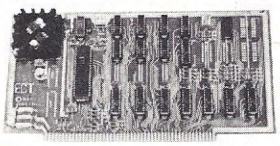
10 SLOT TABLE TOP MICROCOMPUTERS
TT-8080 KIT $475

SYSTEM W/16K & I/O
TT-8080-S KIT $1125



CARD CAGE & MOTHER BOARD
ECT-100 KIT $100

WITH CONNECTORS & GUIDES
ECT-100-F KIT $200



CPU'S, MEMORY
MOTHER BOARDS
PROTOTYPING BOARDS
EXTENDER CARDS
POWER SUPPLIES

DEALER INQUIRIES INVITED *SHIPPING EXTRA*

ELECTRONIC CONTROL TECHNOLOGY
P.O. BOX 6, UNION, NEW JERSEY 07083
(201) 686-8080

CIRCLE 42

TWO BYTES ARE BETTER THAN ONE

COMPARE 16-BIT COMPUTERS

THE TECHNICO SUPER STARTER SYSTEM AND SUPER SYSTEM 16

V.S.

THE HEATH H-11

SAVE OVER $1000

SEND FOR FREE BROCHURE

TECHNICO **1-800-638-2893**
9130 RED BRANCH RD. COLUMBIA, MD 21045
DEALER INQUIRIES INVITED

CIRCLE 44

House-break your PET

When your new PET computer arrives, you and your PET will need to communicate. We have ten (10) programs in the PET's Basic language, on a single quality cassette, ready to run — games, displays, and graphing programs. Use them as is, or modify them to make your own custom programs. Order now so you can house-break you new PET quickly.

Send $19.95 in check or money order to

DON ALAN ENTERPRISES
P.O. Box 401
Marlton, New Jersey 08053
New Jersey Residents Add 5% Sales Tax.
Dealer Inquiries Invited.

CIRCLE 43

**BUSINESSMEN — LAWYERS
DOCTORS — ACCOUNTANTS
WE HAVE THE SYSTEMS YOU WANT & NEED, YOUR 1 SHOP FOR EVERYTHING**

**FULL ACCOUNTING PACKAGE
INVENTORY MANAGEMENT
WORD PROCESSING
DENTAL OFFICE MANAGEMENT
LAW OFFICE PACKAGE**

mits Altair T.M. BUSINESS SYSTEMS

the Computer Store OF NEW YORK

master charge BANKAMERICARD

55 WEST 39th STREET, NEW YORK, NEW YORK 10018
(212) 221-1404

M-T 10:30 — 6 PM SAT. 10:30 — 5 PM

CIRCLE 45


WANTED
IQ 120
DESCRIPTION
Lower Case
Numeric Pad
Protect Mode
Tabbing
Auto Repeat
15 Baud Rates
Dual Intensity
Aux. Port
Addr. Cursor
24 Line x 80 Char.
Optional Printer Port & Block Mode
at $995.00 the
SOROC IQ 120 is the most wanted terminal on the market.
For information concerning the REWARD contact···
SOROC
TECHNOLOGY, INC.
165 FREEDOM AVE., ANAHEIM, CA 92801
714-992-2860 / 800-854-0147

CIRCLE 32

BANNER FUN

BY ED BERNSTEIN

Here's a program that anyone with a computer, a BASIC interpreter, and any hard-copy device could use — a banner program.

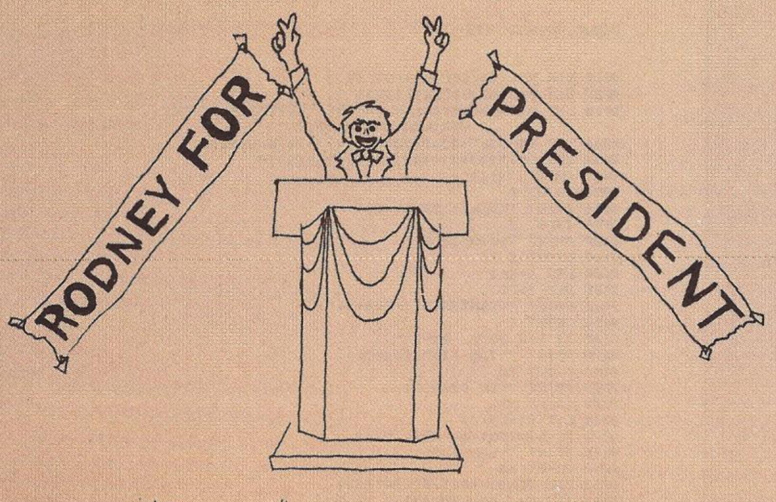
The program prints banners of many different sizes. Actually, it can print an even larger banner, if the output device uses extra-large paper. The program itself has no upper limit.

The characters are 5x7 matrix type, and include, at present, 79 characters. You can print large lower-case letters even if your terminal's keyboard is only upper-case. You can add new characters easily.

The inner characters can be anything on the keyboard, and can be 'offset' any distance. In the examples, the inner characters form diagonal lines, but you can also have columns, diagonals in the other direction, seemingly random placement, or any other pattern you like.

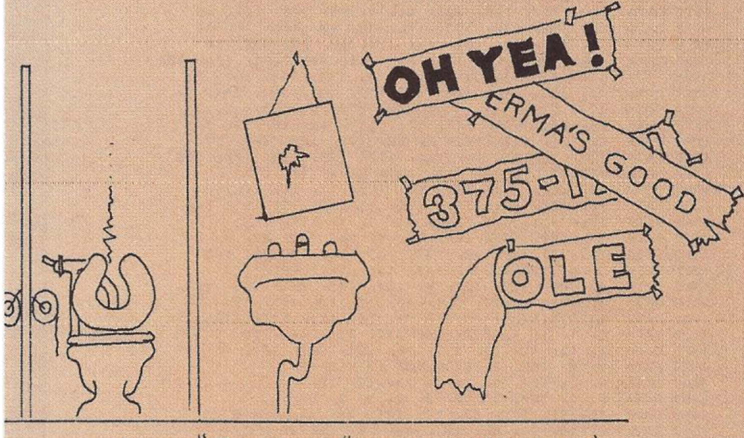
You can make different letters have different sizes in the same banner, and you can place each individual letter wherever you like on the page. The large letters can be "dark" or "light" (a white shape against a dark background).

The banner program does all that. If you want to try changing the program around, then the possibilities are endless. I've written a variation that makes banners half as large as "size one," another that makes letters in a 10x14 typeface designed by a friend, and another that prints letters in Hebrew.



"Banners" make great banners.

The program itself has few limitations. The banner's length, inner characters, and maximum size are limited only by hardware. The minimum size is the size of the matrix used, with the second number multiplied by two, to allow for the difference between vertical and horizontal spacing on a Teletype. I have never seen another banner program that is so versatile and yet so short: only 111 single-statement BASIC lines, and 40 lines of data.



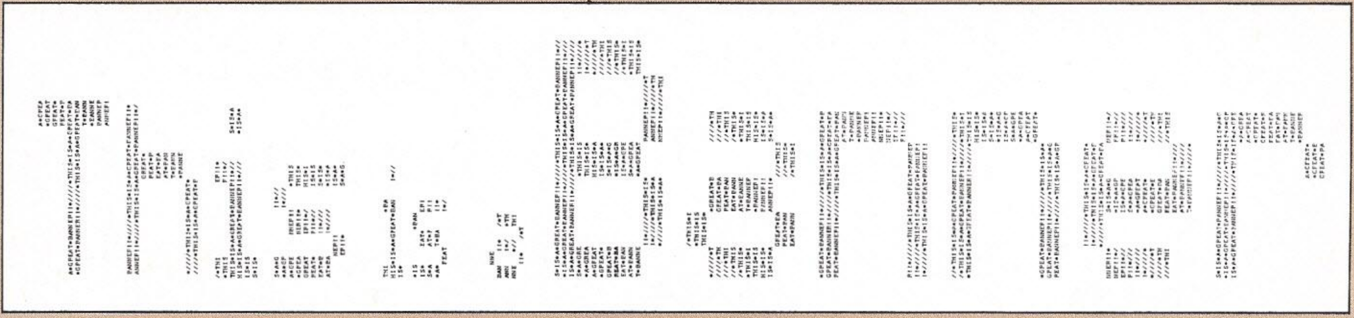
Use "Banners" as creative and non-destructive grafitti.

How to input

The inputs let you choose the exact type of banner that you desire. First, input the "size" (the height of the lower-case letters, in inches), which can range anywhere from 1 to the limit of your printer. (An ASR-33's limit is about 4.9. If you want the size to vary, input 0.

Next, select D or L, meaning Dark or Light.

Then select the offset (how far each successive line of inner characters will be offset to the left). Most banners use an offset of 1, which makes nice diagonal lines out of your inner characters. If the offset is the same as the LEN of your inner characters, then the inner characters will appear in columns. Or try any number that will seem to scramble the characters randomly.

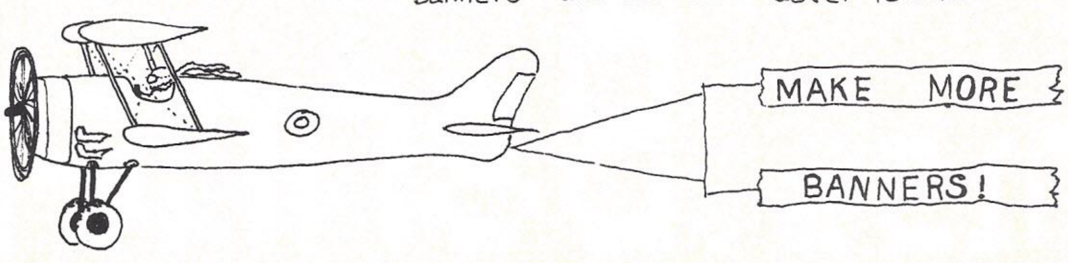


Illustrations by Mark Eike

```
OCTAL BANNER VIB

0010 DIM A$[72],C$[144],D$[69]
0020 DIM B$[80],O$[24],T$[9]
0030 LET B$="ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHI
              JKLMNOPQRSTUVWXYZ"
0040 LET B$=B$,"1234567890.!?-,'$><=/=+:[]()"
0050 LET O$="000001010011100101110111"
0060 PRINT "SIZE";
0070 INPUT D
0080 PRINT "\D(=1) OR L(=2)";
0090 INPUT E
0100 PRINT "\OFFSET";
0110 INPUT C
0120 LET C=C+1
0130 DIM E$[C]
0140 PRINT "\CENTERED(1=Y,0=N)";
0150 INPUT T
0160 IF E=2 GOTO  0190
0170 PRINT "\TAB (1=Y,2=N)";
0180 INPUT T0
0190 PRINT "\IN CHARS";
0200 INPUT A$
0210 LET C$=C$,A$
0220 IF LEN(C$)<69 GOTO 0210
0230 PRINT "\BANNER";
0240 INPUT A$
0250 DIM Q[LEN(A$)],R[LEN(A$)]
0260 FOR X=1 TO LEN(A$)
0270   LET Q[X]=0
0280   LET R[X]=D
0290 NEXT X
0300 IF E=2 GOTO  0430
0310 IF T0<2 GOTO 0330
0320 IF D>=1 GOTO 0430
0330 FOR X=1 TO LEN(A$)
0340   IF A$[X,X]="↑" GOTO  0420
0350   PRINT "\CHAR"X;"("A$[X,X];")";
0360   IF T0>1 GOTO 0390
0370   PRINT " TAB";
0380   INPUT Q[X]
0390   IF D>=1 GOTO 0420
0400   PRINT " SIZE";
0410   INPUT R[X]
0420 NEXT X
0430 PRINT "\\\\\\\\\\\\\\\\"
0440 IF E=1 GOTO  0490
0450 FOR X=1 TO D
0460   GOSUB  1090
0470   PRINT  TAB ((31-D*7)*T);C$[1,D*14+6]
0480 NEXT X
0490 FOR X=1 TO LEN(A$)
0500   IF A$[X,X]="↑" GOTO  1060
0510   LET D=R[X]
0520   LET D$=""
0530   FOR Y=1 TO D*14
0540     LET D$[Y,Y]=" "
0550   NEXT Y
0560   RESTOR
0570   FOR Y=1 TO 79
0580     FOR Z=1 TO 5
0590       READ A[Z]
0600     NEXT Z
0610     IF A$[X,X]<>B$[Y,Y] GOTO 0640
0620     IF A$[X-1,X-1]<>"↑" GOTO 0730
0630     LET A$[X-1,X-1]=" "
0640   NEXT Y
0650   FOR Q1=1 TO D*5
0660     GOSUB  1090
0670     IF E=1 GOTO  0700
0680     PRINT  TAB ((31-D*7)*T);C$[1,D*14+6]
0690     GOTO 0710
0700     PRINT "\";
0710   NEXT Q1
0720   GOTO 1060
0730   FOR B=1 TO 5
0740     LET T$=""
0750     FOR Y=2 TO 0 STEP -1
0760       LET Z=INT(A[B]/10↑Y)
0770       LET T$=T$,O$[Z*3+1,Z*3+3]
0780       LET A[B]=A[B]-Z*10↑Y
0790     NEXT Y
0800     LET T$=T$[3]
0810     IF T$="0000000" GOTO 0980
0820     FOR X1=1 TO D
0830       GOSUB  1090
0840       FOR Y=1 TO 7
0850         LET Z=D*2*(Y-1)+1
0860         IF T$[Y,Y]="1" GOTO  0900
0870         IF E=2 GOTO  0910
0880         LET D$[Z,Z+D*2-1]="             "
0890         GOTO 0920
0900         IF E=2 GOTO  0880
0910         LET D$[Z,Z+D*2-1]=C$[Z+3,Z+D*2+2]
0920       NEXT Y
0930       IF E=2 GOTO  0960
0940       PRINT  TAB ((34-D*7)*T+Q[X]);D$[1,D*14]
0950       GOTO 0970
0960       PRINT  TAB ((31-D*7)*T);C$[1,3];D$;C$[D*14+4,D*14+6]
0970     NEXT X1
0980   NEXT B
0990   FOR Y=1 TO D
1000     GOSUB  1090
1010     IF E=2 GOTO  1040
1020     PRINT "\";
1030     GOTO 1050
1040     PRINT  TAB ((31-D*7)*T);C$[1,D*14+6]
1050   NEXT Y
1060 NEXT X
1070 PRINT "\\\\\\\\\\\\\\\\\\"
1080 END
1090 LET E$=C$[1,C-1]
1100 LET C$=C$[C,LEN(C$)],E$
1110 RETURN
1120 DATA  40, 124, 124, 124, 50, 177, 104, 104, 104, 70
1130 DATA  70, 104, 104, 104, 10, 70, 104, 104, 104, 177
1140 DATA  70, 124, 124, 124, 30, 4, 176, 5, 5, 0
1150 DATA  110, 124, 124, 124, 52, 177, 4, 4, 4, 170
1160 DATA  0, 104, 175, 100, 0, 40, 100, 100, 100, 75
1170 DATA  177, 20, 50, 104, 0, 0, 101, 177, 100, 0
1180 DATA  174, 4, 170, 4, 170, 174, 4, 4, 4, 170
1190 DATA  70, 104, 104, 104, 70, 174, 24, 24, 24, 10
1200 DATA  10, 24, 24, 170, 100, 174, 4, 4, 4, 10
1210 DATA  110, 124, 124, 124, 44, 4, 76, 104, 100, 40
1220 DATA  74, 100, 100, 100, 174, 34, 40, 100, 40, 34
1230 DATA  74, 100, 70, 100, 74, 104, 50, 20, 50, 104
1240 DATA  114, 120, 120, 120, 54, 104, 144, 124, 114, 104
1250 DATA  176, 11, 11, 11, 176, 177, 111, 111, 111, 66
1260 DATA  76, 101, 101, 101, 42, 177, 101, 101, 101, 76
1270 DATA  177, 111, 111, 111, 101, 177, 11, 11, 11, 1
1280 DATA  76, 101, 101, 111, 171, 177, 10, 10, 10, 177
1290 DATA  0, 101, 177, 101, 0, 60, 100, 100, 100, 77
1300 DATA  177, 10, 24, 42, 101, 177, 100, 100, 100, 100
1310 DATA  177, 2, 14, 2, 177, 177, 2, 4, 10, 177
1320 DATA  76, 101, 101, 101, 76, 177, 11, 11, 11, 6
1330 DATA  76, 101, 121, 141, 176, 177, 11, 31, 51, 106
1340 DATA  106, 111, 111, 111, 61, 1, 1, 177, 1, 1
1350 DATA  77, 100, 100, 100, 77, 7, 30, 140, 30, 7
1360 DATA  177, 40, 20, 40, 177, 143, 24, 10, 24, 143
1370 DATA  3, 4, 170, 4, 3, 141, 121, 111, 105, 103
1380 DATA  0, 102, 177, 100, 0, 142, 121, 111, 105, 103
1390 DATA  101, 111, 111, 111, 66, 17, 10, 177, 10, 10
1400 DATA  117, 111, 111, 111, 61, 76, 111, 111, 111, 62
1410 DATA  3, 1, 1, 1, 177, 66, 111, 111, 111, 66
1420 DATA  6, 11, 11, 11, 177, 76, 101, 101, 101, 76
1430 DATA  0, 0, 100, 0, 0, 0, 0, 137, 0, 0
1440 DATA  2, 1, 131, 5, 2, 0, 10, 10, 10, 0
1450 DATA  0, 0, 100, 60, 0, 0, 0, 0, 3, 0
1460 DATA  56, 52, 177, 52, 72, 101, 42, 24, 10, 0
1470 DATA  0, 10, 24, 42, 101, 40, 20, 10, 4, 2
1480 DATA  0, 24, 24, 24, 0, 10, 10, 76, 10, 10
1490 DATA  0, 0, 24, 0, 0, 0, 177, 101, 101, 0
1500 DATA  0, 101, 101, 177, 0, 0, 0, 34, 42, 101
1510 DATA  101, 42, 34, 0, 0
```

"Banners" are low-cost advertisement



"Banners" are great for training pets!

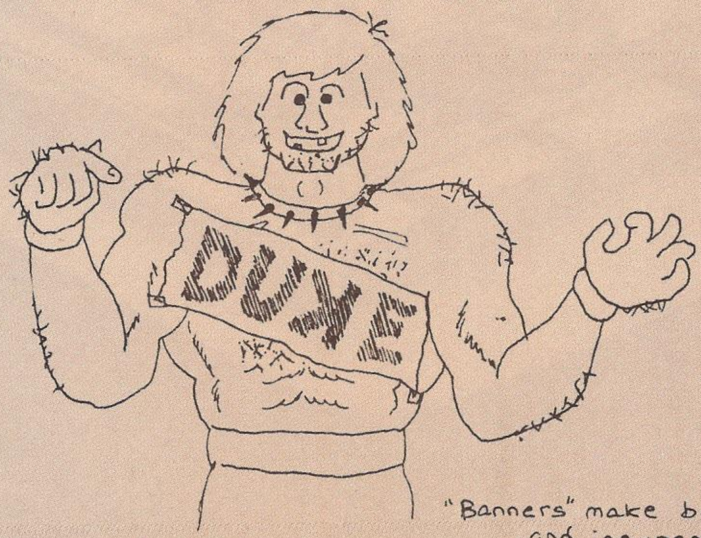
"Centered" will put the banner in the middle of the printed page.

If you said you wanted a dark banner, "Tab" asks whether you want to individually tab the characters in the banner. Answer just yes or no. (The details about tabbing are asked later.)

The next input is the "In char" or inner characters, the small characters that go inside the large banner and make it up. (I haven't put into this program the option to have each letter made up of itself – A's made up of A's, and B's made of B's – because it's ugly!)

After the inner characters, input the banner itself. You can use all the characters on line 40 plus all upper and lower case letters. The letters you type in will normally be printed as lower case. If you want an upper-case letter, type an up-arrow before the letter. (↑HELLO would spell out Hello.)

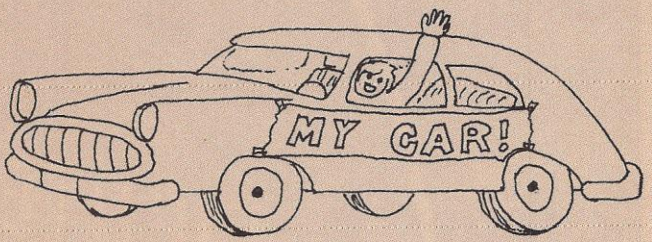
If you said you wanted variable size or tab, then you will have to input more. When the computer asks for the tab and/or size, put in the numbers that you want for the indicated characters. The terminal should then linefeed (a "\" causes a linefeed) and begin printing your banner.



"Banners" make beautiful and inexpensive personalized tee-shirts

Version of BASIC

I wrote the program for a Data General Nova 1200, a computer whose string operations differ from most other computers'. A comma means concatenation between strings (put one string into the other). A$[X,Y] references A$, from character number X (counting from the left, the first character is one) through character Y. If you want only one character, it would be something like A$(x,X). The beauty of this system is that this substring can be assigned a value (LET A$(X,Y) = "HI!"), just like any other independent



Use "Banners" to Personalize your possesions.

variable. I leave this program to you to translate into your own version of BASIC.

Data

Since each letter is formed out of a 5x7 matrix, each letter is represented by 5 numbers in the DATA statement. Since the data is in base eight (octal), a three-digit number can represent up to 9 bits, but I need only 7. Here is an example:

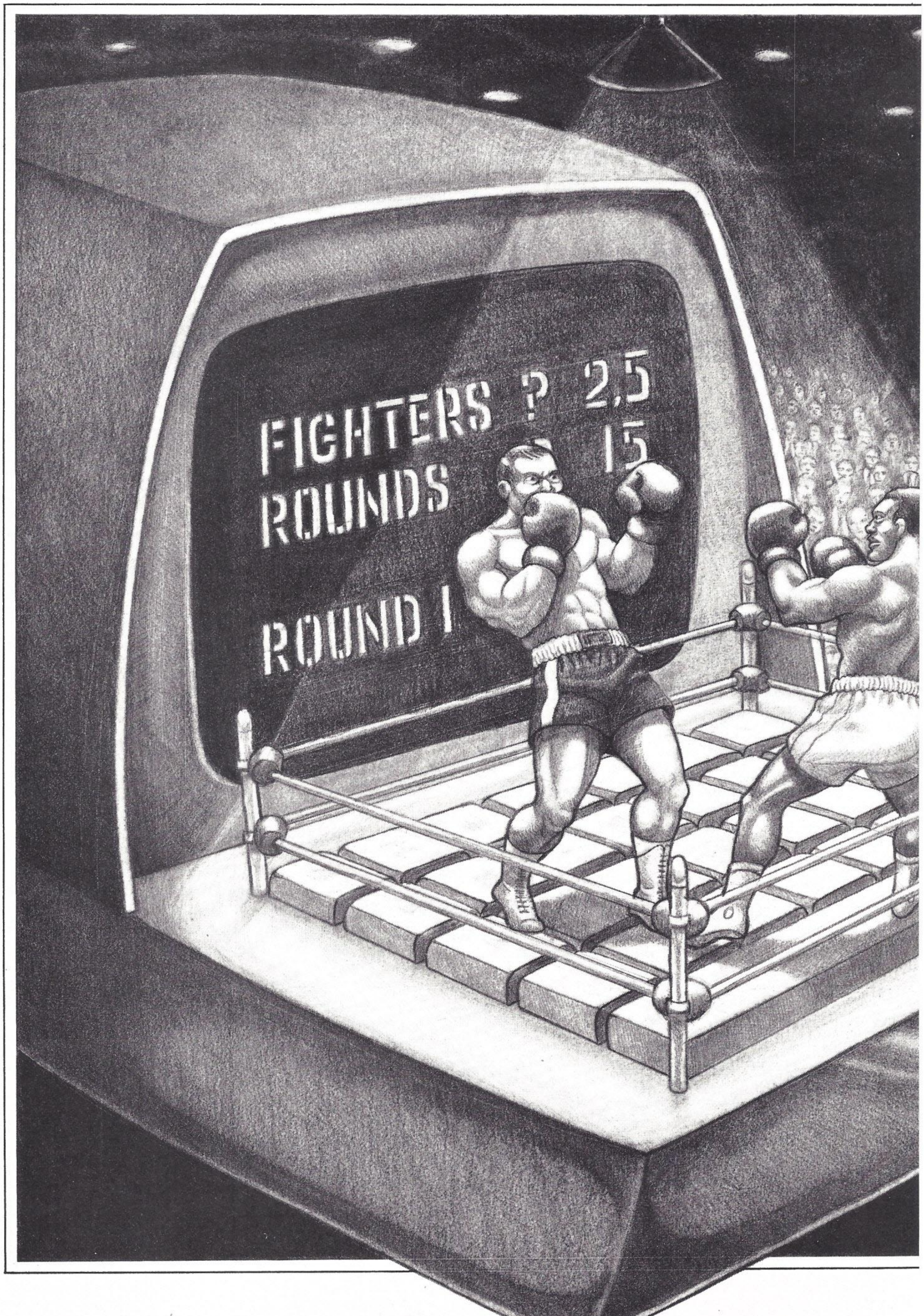
CODING THE LETTER A (READ SIDEWAYS)

OCTAL	BINARY							
176	1111110	●	●	●	●	●	●	
011	0001001				●			●
011	0001001				●			●
011	0001001				●			●
176	1111110	●	●	●	●	●	●	

The computer reads data into array A. If that data is not for the correct letter, the computer discards that data and reads different data instead. Once the computer finds the correct data, it converts each octal number into a string of one's and zero's. The string indicates where the inner characters should go.

Designing letters for the banner program is easy. Lowercase letters are in lines 1120-1240, upper case in 1250-1370, numbers in 1380-1420, and the other characters come afterwards. To add characters, put the data after 1510, put the characters at the end of B$, increase the dimension of B$ accordingly, and change line 570 to FOR X=1 TO LEN (B$).

Since the banner program is so versatile, half the fun is creating new characters and different letter styles. The banners lend themselves to many varied uses – not all of which are immediately apparent. PC


FIGHTERS ? 2,5
ROUNDS ? 15
ROUND 1

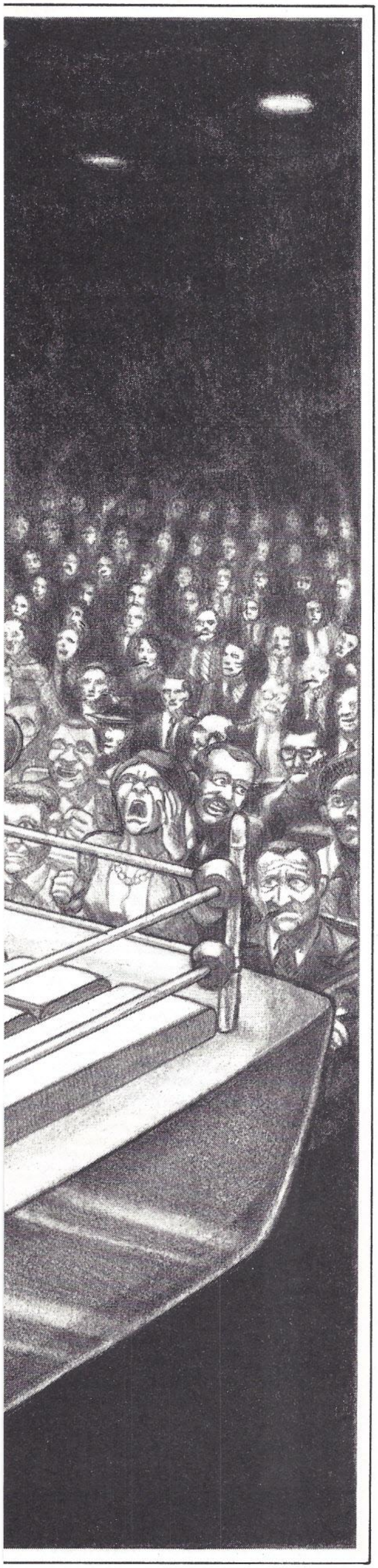


Illustration by Douglas Smith

BOXING BY COMPUTER

BY JOSEPH J. ROEHRIG

This program demonstrates how random numbers coupled with preset values can simulate a boxing match. Suppose you are at a video terminal and the program is already loaded into memory. Just type "RUN" as shown in Figure 1 and enter a random number ("3" in the example).

Then the computer will give you three choices. You can enter "0" to end the program, "1" for a list of the fighters or "2" to choose fighters for a bout. In the sample run, "1" is selected, so the computer lists twenty fighters. The list gave the fighters' rank, bouts won and lost, KO's (knockouts or technical knockouts) and the fighters' identification number. Then the computer gives you three choices again.

This time a "2" is entered, so a fight will be staged. The program selects the fighters for the match and in response two identification numbers are entered. For Figure 1, "1" and "12" were entered as the identification numbers. Next you select the length of the fight by inputting the rounds ("15" for this event).

Now the computer is ready to proceed with the simulation. The fighters' names are printed "ALI VERSUS WEPNER" (corresponding to identifications 1 and 12) and the fight is on (Figure 2). The printout gives the round-by-round action and at the end of each round presents the scoring, represented by the two numbers on the righthand side of the printout. The first number is the first fighter's points, and the second number is the second fighter's. For the first round of the example in Figure 2, Ali scored 5 points to Wepner's 1.

Our sample fight has Ali stopping Wepner (known as the Bayonne Bleeder) on cuts in the second round. Wepner was over-matched. Figure 3 shows the legendary Rocky Marciano stopping Ken Norton in five rounds. To show a fight coming to a decision (Figure 4), Ali was matched against a tougher opponent – Jimmy Young, who in real life fought a 15-round bout with Ali and lost a close decision to Ali – for only five rounds. In this contest, we see Ali beating Young three rounds to two. The scoring is on a round basis. So even though Young scored the same 22 points as Ali scored, Young lost because he won less rounds.

Now let's go to our "LIST" via a "1" entry. Figure 5 shows the updated rankings and records after the three matches had been fought. The ranking always ranks an active fighter (one who fought in a match during the program's execution) higher than an inactive fighter. The only way a fighter moves up in the rankings is to beat a fighter whose ranking is higher.

The simulation runs on an IMSAI

FIGURE 1 – RUN

```
RANDOM ? 3
0-END 1-LIST 2-FIGHT ? 1

RANK NAME      WINS LOSS DRAW KO'S ID
-------------------------------------
  1 ALI          0    0    0    0  1
  2 MARCIANO     0    0    0    0  2
  3 FRAZIER      0    0    0    0  3
  4 FOREMAN      0    0    0    0  4
  5 NORTON       0    0    0    0  5
  6 YOUNG        0    0    0    0  6
  7 PATTERSO     0    0    0    0  7
  8 LISTON       0    0    0    0  8
  9 JOHNASON     0    0    0    0  9
 10 SHAVERS      0    0    0    0 10
 11 BOBICK       0    0    0    0 11
 12 WEPNER       0    0    0    0 12
 13 QUARRY       0    0    0    0 13
 14 LYLE         0    0    0    0 14
 15 BUGNER       0    0    0    0 15
 16 ELLIS        0    0    0    0 16
 17 ROMAN        0    0    0    0 17
 18 BLUE LOU     0    0    0    0 18
 19 CHAVALO      0    0    0    0 19
 20 AGOSTO       0    0    0    0 20
0-END 1-LIST 2-FIGHT ? 2
FIGHTERS ? 1,12
ROUNDS ? 15
```

FIGURE 2

```
ALI          VERUS   WEPNER

ROUND 1     WEPNER   BODY PUNCHES                ALI
            ALI      UPPERCUTS                   WEPNER
            ALI      JABS                        WEPNER
            ALI      BODY PUNCHES                WEPNER
            ALI      UPPERCUTS                   WEPNER
                                               5 1

ROUND 2     WEPNER   LANDS TO THE HEAD OF        ALI
            ALI      LANDS TO THE HEAD OF        WEPNER
            ALI      SCORES TO THE BODY OF       WEPNER
            ALI      BODY PUNCHES                WEPNER
            ALI      SCORES TO THE BODY OF       WEPNER
            ALI      OPENS A BIG CUT ON          WEPNER
            ALI      JABS                        WEPNER
            ALI      UPPERCUTS                   WEPNER
            THE REF STOPS IT ALI          WINS
0-END 1-LIST 2-FIGHT ?
FIGHTERS ? 2,5
ROUNDS ? 15
```

FIGURE 3

```
MARCIANO   VERUS   NORTON

ROUND 1    NORTON     BODY PUNCHES                 MARCIANO
           NORTON     UPPERCUTS                    MARCIANO
           MARCIANO   THROWS A LEFT DOWN GOES      NORTON
                                                 5 4

ROUND 2    MARCIANO   SCORES TO THE BODY OF        NORTON
           MARCIANO   LANDS TO THE HEAD OF         NORTON
           NORTON     UPPERCUTS                    MARCIANO
           NORTON     SCORES TO THE BODY OF        MARCIANO
           NORTON     SCORES TO THE BODY OF        MARCIANO
                                                 3 4

ROUND 3    NORTON     SCORES TO THE BODY OF        MARCIANO
           MARCIANO   UPPERCUTS                    NORTON
                                                 2 1

ROUND 4    NORTON     JABS                         MARCIANO
           MARCIANO   SCORES TO THE BODY OF        NORTON
           NORTON     LANDS TO THE HEAD OF         MARCIANO
                                                 1 4

ROUND 5    MARCIANO   HITS HARD AND STAGGERS       NORTON
           MARCIANO   BODY PUNCHES                 NORTON
           MARCIANO   OPENS A BIG CUT ON           NORTON
           MARCIANO   THROWS A LEFT DOWN GOES      NORTON
           THE REF STOPS IT MARCIANO WINS
0-END 1-LIST 2-FIGHT ? 2
FIGHTERS ? 1,6
ROUNDS ? 5
```

FIGURE 4

```
ALI         VERUS   YOUNG

ROUND 1    YOUNG     JABS                        ALI
           YOUNG     SCORES TO THE BODY OF       ALI
           ALI       SCORES TO THE BODY OF       YOUNG
           ALI       UPPERCUTS                   YOUNG
           YOUNG     JABS                        ALI
           YOUNG     LANDS TO THE HEAD OF        ALI
                                               2 6

ROUND 2    ALI       SCORES TO THE BODY OF       YOUNG
           ALI       SCORES TO THE BODY OF       YOUNG
                                               2 0

ROUND 3    YOUNG     BODY PUNCHES                ALI
           ALI       LANDS TO THE HEAD OF        YOUNG
           ALI       UPPERCUTS                   YOUNG
           ALI       LANDS TO THE HEAD OF        YOUNG
           YOUNG     JABS                        ALI
           ALI       OPENS A BIG CUT ON          YOUNG
                                               9 4

ROUND 4    YOUNG     BODY PUNCHES                ALI
           YOUNG     UPPERCUTS                   ALI
           YOUNG     BODY PUNCHES                ALI
           YOUNG     JABS                        ALI
           ALI       UPPERCUTS                   YOUNG
           ALI       BODY PUNCHES                YOUNG
           YOUNG     BODY PUNCHES                ALI
                                               3 7

ROUND 5    YOUNG     SCORES TO THE BODY OF       ALI
           ALI       JABS                        YOUNG
           ALI       BODY PUNCHES                YOUNG
           ALI       BODY PUNCHES                YOUNG
           ALI       SCORES TO THE BODY OF       YOUNG
           YOUNG     LANDS TO THE HEAD OF        ALI
           YOUNG     BODY PUNCHES                ALI
                                               6 5

THE DECISION

ALI       3 ROUNDS --   YOUNG      2 ROUNDS
```

8080, using an ADM-3 video terminal and North Star BASIC. To control the fighters' performances, you assign eleven ratings to each fighter. Ali's ratings were:

RATING	VALUE	REPRESENTS
1	8	ability to maintain offense
2	12	ability to maintain defense
3	14	offensive ability to score 1-point blows
4	21	offensive ability to score 2-point blows
5	25	offensive ability to cut an opponent
6	28	offensive ability to stagger or injure an opponent
7	31	offensive ability to knock down an opponent
8	10	defensive skills to avoid cuts
9	5	defensive skills to avoid staggering and injury
10	5	defensive skills to avoid being knocked down
11	20	defensive skills to avoid being knocked out

The actual simulation dissected every round into twenty possible action situations and generated a random number between 1 and 100 for each. A random number at the start chose one fighter as being on the attack. When Ali is on the attack, a random number is compared to Ali's five offensive ratings. Ratings 3 to 7 are compared in sequence against the random number. The first rating greater than the random number determines the offensive option, and a random number higher than any of the offensive ratings transfers the attack mode to the opponent. If the rating is 5, 6, or 7, the program generates a second random number and compares it to the opponent's corresponding defensive rating. If the sec-

FIGURE 5 – NEW RANKS

```
0-END 1-LIST 2-FIGHT ? 1

RANK NAME      WINS LOSS DRAW KO'S ID
--------------------------------------
   1 ALI         2    0    0    1  1
   2 MARCIANO    1    0    0    1  2
   3 NORTON      0    1    0    0  5
   4 YOUNG       0    1    0    0  6
   5 WEPNER      0    1    0    0 12
   6 FRAZIER     0    0    0    0  3
   7 FOREMAN     0    0    0    0  4
   8 PATTERSO    0    0    0    0  7
   9 LISTON      0    0    0    0  8
  10 JOHNASON    0    0    0    0  9
  11 SHAVERS     0    0    0    0 10
  12 BOBICK      0    0    0    0 11
  13 QUARRY      0    0    0    0 13
  14 LYLE        0    0    0    0 14
  15 BUGNER      0    0    0    0 15
  16 ELLIS       0    0    0    0 16
  17 ROMAN       0    0    0    0 17
  18 BLUE LOU    0    0    0    0 18
  19 CHAVALO     0    0    0    0 19
  20 AGOSTO      0    0    0    0 20
0-END 1-LIST 2-FIGHT ? 0
READY
```

ond random number is lower than the defensive rating, the offensive move occurs; if it is higher, the opponent's defense prevents the offensive advance.

If during any round a point value higher than a fighter's #1 rating is scored against him, all his offensive skill is reduced by 1, and if the fighter's #2 rating is bettered, all of the defensive ratings are increased by 1.

Now that you're familiar with the ratings, you can re-rate the fighters or add fighters. But note that the four offensive skill ratings (4 to 7) must always be higher than the preceding rating or the event will never occur.

Figure 6 shows the key variables and line number functions, while Figure 7 is the complete listing.

FIGURE 6 – DECIPHERING THE PROGRAM

KEY VARIABLES

F(20,11)	- 20 fighters and 11 ratings
S(20,5)	- rank, wins, losses, draws, KO's and rating points for the boxers
N$(160)	- the names of fighters
Z(2,11)	- ratings for the two fighters in a match (since the rating can be changed by the condition occurring in the bout and the initial data must be maintained)
R(2)	- rounds won
P(2)	- points scored in a round
W$	- descriptive output
G$	- descriptive output

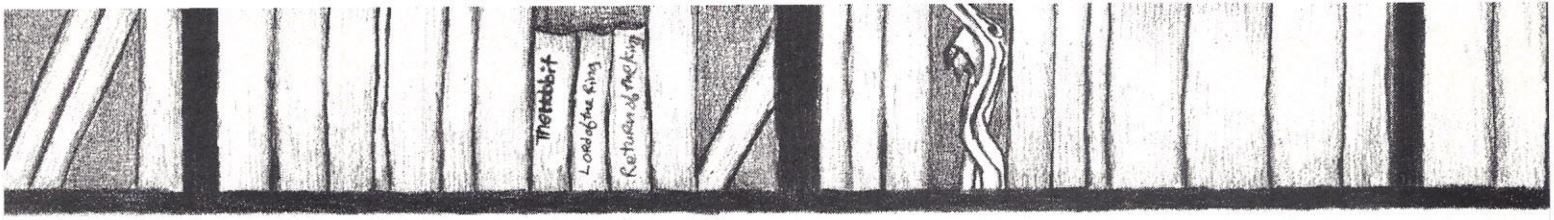
LINE NUMBERS

1-80	- dimensions variables and reads data
90-100	- asks user for action code
110-210	- prints lists
220-560	- the fight
570-720	- updates boxers' records
750-770	- additional descriptions for the fight
1000-1008	- descriptive data
1101-1120	- data for the 20 fighters

FIGURE 7 – PROGRAM LISTING

```
10 DIMF(20,11),S(20,5),N$(160),Z(2,11),R(2),P(2),
     W$(115),G$(25)
15 FORA=1TO5\B=A*23\READW$(B-22,B)\NEXT
17 G$='BODY PUNCHESJABSUPPERCUTS'
20 FORA=1TO20
30 B=A*8
40 READN$(B-7,B)
50 FORB=1TO11
60 READF(A,B)\NEXTB\NEXTA
80 INPUT'RANDOM ? ',A\B=RND(.5)\FORB=1TOA\C=
     RND(0)\NEXT
90 INPUT'0-END 1-LIST 2-FIGHT ? ',A
95 IFA=0THENEND
100 IFA>1THEN220
110 FORA=1TO20\S(A,0)=S(A,5)\NEXT
120 !''\  !'RANK NAME      WINS LOSS DRAW KO'S ID'
130 PRINT'------------------------------------------'
140 FOR A=1TO20\B=0-999
150 FORC=1TO20
160 IFS(C,0)<=BTHEN175
170 B=S(C,0)\D=C
175 B=B
180 NEXT
190 S(D,0)=0-9999
195 V2=D*8\V1=V2-7
200 PRINT%4I,A,' ',N$(V1,V2),%5I,S(D,1),S(D,2),
     S(D,3),S(D,4),%3I,D
210 NEXT A\GOTO90
220 INPUT'FIGHTERS ? ',A1,A2
230 IFA1>20THEN220\IF1>A1THEN220
240 IFA2>20THEN220\IF1>A2THEN220
245 S(A1,5)=S(A1,5)+.001\S(A2,5)=S(A2,5)+.001
250 FORC=1TO11
255 Z(1,C)=F(A1,C)\Z(2,C)=F(A2,C)\NEXT
260 INPUT'ROUNDS ? ',A\R(1)=0\R(2)=0\V1=A1*8\V2=A2*8
262 !''\!N$(V1-7,V1),'  VERUS  ',N$(V2-7,V2)\!''
265 M=1\M1=2
270 FOR B=1 TO A\PRINT' '
280 PRINT'ROUND',B,
290 P(1)=0\P(2)=0
295 FOR C= 1 TO 20
300 M2=INT(RND(0)*100)+1
310 FORD=3TO7
320 IFZ(M,D)=>M2THENEXIT340
330 NEXTD\GOTO520
340 IF5>DTHEN750
348 FORK=1TOD
350 M2=INT(RND(0)*100)+1
351 NEXT K
360 IFM2>Z(M1,D+3)THENGOTO530
370 M3=((D-3)*23)+1
380 PRINTTAB(10),N$(V1-7,V1),' ',W$(M3,M3+22),' ',
     N$(V2-7,V2)
390 P(M)=P(M)+D-2
400 IFD<>7THEN450
410 M2=INT(RND(0)*100)+1
420 IFM2>Z(M1,10)THEN450
430 PRINTTAB(10),'IT'S OVER ',N$(V1-7,V1),
     ' WINS ON A KNOCKOUT'
440 GOTO700
450 IFZ(M1,11)>P(M)THEN470
460 PRINTTAB(10),'THE REF STOPS IT ',N$(V1-7,V1),
     ' WINS'\GOTO700
470 IFZ(M1,1)>P(M)THEN500
490 FORE=3TO7\Z(M1,E)=Z(M1,E)-E+2\NEXT
500 IF Z(M1,2)>P(M)THEN530
510 FORE=8TO10\Z(M1,E)=Z(M1,E)+E-7\NEXTE\GOTO 530
520 M9=M1\M1=M\M=M9\M9=V1\V1=V2\V2=M9
530 NEXT C
540 IFP(1)>P(2)THENR(1)=R(1)+1
550 IFP(2)>P(1)THENR(2)=R(2)+1
555 !TAB(40),P(1),P(2)
557 P(1)=0\P(2)=0
560 NEXT B
570 PRINT'THE DECISION'\!' '\!' '
572 V1=A1*8\V2=A2*8
575 !N$(V1-7,V1),R(1),' ROUNDS --  ',N$(V2-7,V2),R(2),' ROUNDS'
580 IF R(1)<>R(2)THEN650
590 PRINT'IT'S A DRAW'
595 S(A1,3)=S(A1,3)+1\S(A2,3)=S(A2,3)+1
600 IFS(A1,5)>S(A2,5)THENS(A2,5)=S(A1,5)-.0001
610 IFS(A2,5)>S(A1,5)THENS(A1,5)=S(A2,5)-.0001
620 GOTO 90
650 M=A1\M1=A2\IFR(1)>R(2)THEN670
660 M=A2\M1=A1\V1=V2
670 S(M,1)=S(M,1)+1\S(M1,2)=S(M1,2)+1
675 IFR(1)>R(2)THENR(2)=R(1)
676 IFR(2)=99THENS(M,4)=S(M,4)+1
680 GOTO 710
700 R(M)=99\GOTO650
710 IFS(M,5)<S(M1,5)THENS(M,5)=S(M1,5)+1
715 S(M,5)=S(M,5)+.01
720 GOTO 90
750 M2=INT(RND(0)*5)+1
760 ONM2GOTO370,370,762,764,766
762 M3=1\M4=12\GOTO768
764 M3=13\M4=16\GOTO768
766 M3=17\M4=25
768 !TAB(10),N$(V1-7,V1),' ',G$(M3,M4),' ',
769 !TAB(43),N$(V2-7,V2)
770 GOTO 390
1000 DATA'SCORES TO THE BODY OF    '
1002 DATA'LANDS TO THE HEAD OF     '
1004 DATA'OPENS A BIG CUT ON       '
1006 DATA'HITS HARD AND STAGGERS   '
1008 DATA'THROWS A LEFT DOWN GOES '
1101 DATA'ALI     ',8,12,14,21,25,28,31,10,5,5,20
1102 DATA'MARCIANO',12,9,4,8,10,16,22,20,10,5,20
1103 DATA'FRAZIER ',10,7,6,7,9,13,17,40,30,15,12
1104 DATA'FOREMAN ',8,6,3,6,7,14,22,20,40,40,20
1105 DATA'NORTON  ',8,6,10,20,22,24,26,10,20,30,14
1106 DATA'YOUNG   ',12,12,12,24,24,24,24,20,20,10,20
1107 DATA'PATTERSO',7,7,7,14,15,16,20,10,30,80,16
1108 DATA'LISTON  ',10,10,1,2,3,10,20,20,20,50,18
1109 DATA'JOHNASON',6,6,2,4,10,16,22,50,50,50,12
1110 DATA'SHAVERS ',8,5,2,4,4,9,16,30,50,50,14
1111 DATA'BOBICK  ',8,8,6,12,14,18,20,20,20,20,15
1112 DATA'WEPNER  ',6,6,3,6,9,12,14,90,50,45,10
1113 DATA'QUARRY  ',7,7,4,8,10,13,16,80,40,40,10
1114 DATA'LYLE    ',2,2,1,2,10,15,20,40,40,40,20
1115 DATA'BUGNER  ',9,9,6,9,10,11,12,30,10,20,9
1116 DATA'ELLIS   ',8,8,4,7,8,9,10,20,20,30,12
1117 DATA'ROMAN   ',4,4,6,7,8,9,10,10,80,90,20
1118 DATA'BLUE LOU',3,3,2,4,6,8,10,50,50,50,12
1119 DATA'CHAVALO ',16,16,3,6,9,12,14,70,40,5,20
1120 DATA'AGOSTO  ',5,5,2,4,6,8,10,30,60,60,8
```


BEST of
MAD
POPULAR Mechanics
GREAT
Expectations
BEOWULF



Elegy to a four place slide rule

This recollection, written on a day in summer when temperatures went to 96° F, includes some facts that appear to have frizzled under a hot sun.

BY HARRY SHERSHOW

Hot winds of eternity flow constantly across the face of Earth depositing seeds of pure cosmic dust that grow into unusual creatures. If we did not know we were human beings moving about and taking pictures of each other, it would be hard to believe that anything as fantastic as ourselves could possibly exist. The only real function we have is an ability to create temporary things more meaningless than ourselves, like disappearing ink. We do not know why we are here. We are mud pies turning out more mud pies. Not typical of what nature usually constructs, we must consider ourselves mistakes. If we were assembled from cloth and buttons we would be found in bargain basements marked "irregulars." Instead of eating insects, flying, laying eggs, and digging holes in the ground, we pick our teeth, comb our hair, shine our shoes and wear neckties. We were probably cast upon the ground as seeds to be eaten by other creatures, just as we throw corn to chickens so they can grow fat for markets. But we're lucky enough to sprout up before we get eaten.

An astronomer once said to me, "You see all those stars, moons and milky ways up in the skies, kid?"

"Yeah," I said.

"Did you know that the *whole* mingle-mangle is rushing through space at the speed of light?"

"No."

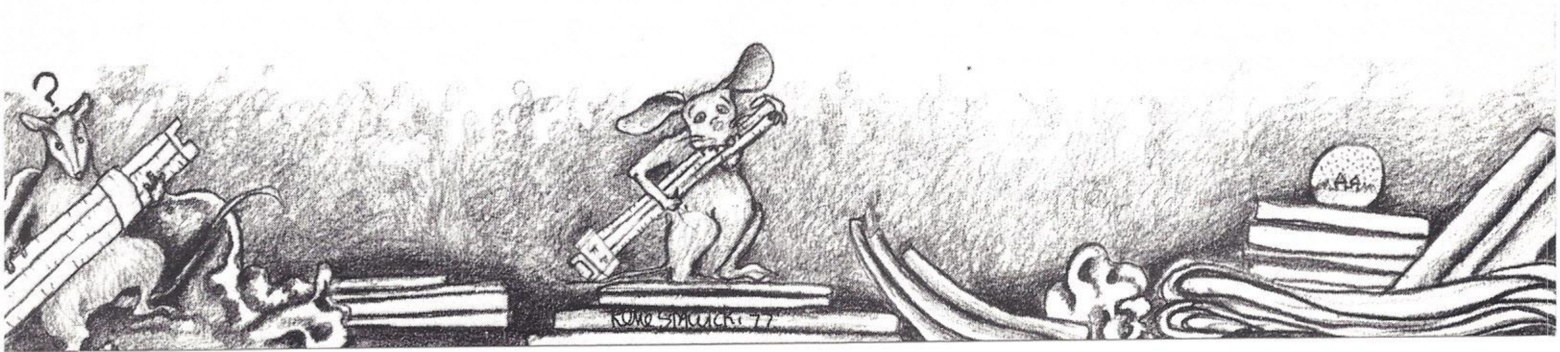
"Why do you suppose they're in such a big hurry?"

"I don't know."

"Neither do I."

So when a pipsqueak like me goes through a gigantic thing like life, I'm beaten black and blue and suffer pains and headaches. It's probably a good thing. Makes you realize that life is real and no monkey business.

If you flew through the air at all times, just a few feet above ground, with eyes half closed in contentment and feeling no pain, no recollections, no desires, no temptations, no hunger, life would have no substance then! Might just as well be born asleep and continue to snooze all the way to the grave.



So life reveals its design only when it is mixed up the way it is and offers continual dramas of strange events.

And the noise! My God! The clamor we generate from this planet! Probably the noisiest satellite in the whole silent cosmic system. Everyone tip-toeing around space except us. I'm sure some quiet guy on one of the further planets is listening to us with headphones hooked into a big radar dish. And he must be dumbfounded by our constant telephone conversations, our unending television gabbing, the deafening public address systems, the thunderous roars from sporting events, and multitudes of other sound waves

The first documented existence of a computer was a punched-card gadget introduced briefly in 1830 but quietly abandoned the following year because it had no practical purpose.

bombarding him. Worst of all, everyone arguing. Not a kind word exchanged between two people on Earth for any length of time. Everyone yapping! And the radar-listener keyed onto us must say to himself, "What in hell are they arguing about?"

"Quiet, down there!" he yells. "Say nice things to each other!"

But no one hears him. Everybody's too busy arguing. Perhaps the government should declare one day a year as "Nice Day." Talk gently to each other, on that day, and say only nice things. Anybody that argues gets thrown into Levenworth.

Arguing is nothing new for us. We've been doing it for the past 10,000 years – ever since our voices emerged from the chirping stage. And of all fields of human controversy, science has had more than its share. Scientific arguments date further back in annals of antiquity than other squabbles. Man's earliest primitive controversy, which raged 10,000 years ago, was the "leaf-God" doctrine. Some cavemen preached that when a wind-blown leaf tumbled mysteriously across open fields it was the sign of an angry God demanding to be worshipped. Other cavemen disagreed: the leaf was a devil, they argued, and they clubbed it to death whenever they saw it moving. This put everyone into a terrible frenzy and led to development of military science, tribal wars and premature ejaculations.

Other ancient controversies concerned flying, rotation of planets, shape of the Earth and origin of disease. Early physicians, for instance, postulated that disease was caused by demons dwelling inside the human body. Popular treatment at the time was the drilling of holes in skulls to let out the demons. In those days, a dozen tough men were needed to drag an ailing friend to the office of a skilled doctor. This is probably the origin of the primitive human instinct that makes people reluctant to visit their doctors.

A controversy of more recent vintage, the one that involved Professor von Karlstein's slide rule research versus computers, ended in 1939 after a strange series of events, which I happened to observe, and deserves to be reviewed at this time.

Scientists and science students split into two groups in the 1930's. On the one hand, those of us who used the slide rule praised it fervently while others, like von Karlstein, sought to make it more versatile. John Napier (if you want to trace things back to the original source) was the guy that really started it all. In 1614 he figured out a system he called logarithms in which, instead of multiplying you added and instead of dividing you subtracted. He prepared a log table in which numbers from 1 to 100 were given logarithmic equivalents. To multiply 2 x 3 you simply took the log of 2 (3010) and added it to the log of 3 (4772) and you got 7782 as an answer. In the log table you looked up the number 7782 and you found it came to 6, the product of multiplying 2 x 3.

An arithmetic student of the 6th grade who lived next door to John Napier, the inventor, stopped him on the street one day and said, "Mister Napier! Why don't you just use your fingers?" He held up 3 fingers of each hand to show that two 3's equal six fingers, which was the way they multiplied things at school.

"Shaddap!" said John Napier. "Mind your own business!"

So logarithms were born. Shortly afterward somebody else doodled around with the 12" ruler, cut a groove into it (spoiling a perfectly good measuring instrument), added a sliding center, erased the inch markings and put in logarithmic graduations. For 300 years after this birth of the slide rule, it was the unchallenged leader in the field of calculation devices. Sticking out of a shirt pocket, it became the familiar symbol of the practicing engineer. Just shows how we get things done here on Earth.

Computer people, on the other hand, have been around only 40 years or so. The computer itself has been around longer than that, though no one paid any attention to it. The first documented existence of a computer was a punched-card gadget introduced briefly in 1830 but quietly abandoned the following year because it had no practical purpose. (Someone had invented the cart before the horse.) In 1872, Lord Kelvin constructed a copper-drum computer that worked like a player piano and was designed to predict tides. After two weeks of forecasting tides for the following two and a half centuries, the gadget, having served its only purpose, was scrapped.

By the 1930's, Professor von Karlstein, German-trained mathematician, was acknowledged leader of the slide group. He warned against using the newly-designed computers because they depended solely on the whims of resistors, amplifiers and capacitors. At any moment, the professor used to argue, a resistor might burn out and a long series of computations would be lost in the computer. Slide rules, on the other hand, were completely reliable and fireproof. In his entire adult lifetime, Heinrich von Karlstein treated the slide rule with the same affection that a colonel accords a beloved swagger stick. I myself have seen him kiss it goodnight just before putting it in his pocket. In lecture halls the professor used it as a blackboard pointer or would slap it soundly into an open palm anytime he got mad at somebody. In the street, he tucked it securely in his armpit, protected from the elements, the way a colonel tucks his swagger stick under the arm.

His teaching career began at Heidelberg as instructor of hydraulics. I learned this in his classes – he used to reminisce a lot between problems. His keen, restless mind drew him into exotic fields of research. The first paper he ever submitted, for example, explored the enormous hydraulic pressures in the circulatory system of whales. You can find a copy of this paper in the MIT library. The paper, handwritten in German, was never published. Using triangular plottings of a diving whale, he had concluded that the reason whales pee through the top of their heads was due to high hydraulic pressures in renal arteries. He didn't actually use the word "pee." He wrote "Wasserverglessen," but someone had made a note in the margin in red ink: "Pee!"

At any moment, the professor argued, a resistor might burn out and a long series of computations would be lost in the computer.

The paper marked the professor as an eccentric scholar whose future was worth watching. Those were the words written on the margin by the same man with the red ink. "Eccentric! Watch him!" were the comments.

He had been a brilliant mathematician in his native Hamburg and nobody played with him. Nobody likes a wise kid, especially other kids. When only 13 years old, he already had solved quadratic equations, squared the circle, trisected an angle and proved that all straight lines were gently curved around infinity. For all that, Albrecht Stutzfein, perhaps the dumbest kid in school, gave him a black eye.

Irrepressible Heinrich's energy was undiminished. From an old piano he constructed an accurate mahogany slide rule that not only performed multiplication and division but also played the "Tannenbaum" melody as the tongue slid back and forth in its groove. Practical relatives of the von Karlstein family admired the handsome mahogany slide rule and predicted that, with a few lickings to put him on the right path, Heinrich would someday become a great, successful cabinet maker. But he ignored family pressures, which were pulling him in different directions, and he drifted into the field of science where his mind could blossom out. (All this is recorded in the "Engineering Chronicles," which gives the autobiographies of well-known scientists.)

As a young man maturing in a cosmopolitan German society, Heinrich had no urbanity, no manners. He picked his nose in public, wore fancy argyle socks under heavy work boots and always sported the same, stained, red-and-green Christmas necktie hanging carelessly over the left shoulder.

He was short, stocky, red-haired with large head, freckled face, bulbous nose (inherited from his grandfather, who drank too much) and small piercing eyes. He had a deep, serious, clenched-fist type of arrogance. Just to show how arrogant he was, if you stopped him on the street and asked him what time it was, he would come to a sharp halt, click his heels, stiffen himself to full military attention, and sim-

ply ignore you. He would glance at his watch but he wouldn't tell you what time it was. He was too arrogant. His philosophy was: if you wanted to know what time it was that bad, *you* buy a watch.

In his growing-up-years, most embarrassing of all were those times he went to German dance halls to do the "Eins! Zvei! Drei!" This brilliant mathematician always lost count during the Ausflug. He would get as far as "Zvei!" and then stamp down hard on his partner's toes. The girls stopped dancing with him and pushed him further outside their social world.

There was a fleeting moment in Heinrich's life when he almost made it back to society. He frequently went to a Hamburg Bierhalle on the shores of the Elbe River where he sat with beer stein in hand and watched the water flow by on its way to the Baltic Sea. There he met and became extremely fond of a nice ländlich Fraulein, a down-to-earth girl. She was the homey type his family fell in love with. He was encouraged to pursue the romance, because the family felt she'd make a good hausefrau and make the difference in Heinrich's life. She was a muscular woman, with deep baritone voice and a well-paying job laying bricks in construction. One night, while they were slobbering over some Wurtzberger, and the Bierhalle lights were turned down low, and the strolling violin player was playing German Lieder, she asked Heinrich to marry her. Heinrich said yes even before she said please. The wedding, quickly performed that same night, lasted only until the following morning, when Heinrich sobered up to find his wife humming a song in the bathroom and shaving her face with a straight razor. The incident increased his bitterness toward society which, he figured, was determined to play dirty tricks on him. He never again had anything to do with girls, but confined himself completely to his work.

The brilliant mathematician always lost count during the Ausflug. He would get as far as "Zvei!" and then stamp down hard on his partner's toes.

He arrived in this country in 1932 and obtained a low-paying instructorship at MIT. Within a year he was awarded a small regular monthly grant from Keuffel and Esser Slide Rule Company in Germany. The money helped him continue earlier research on the slide rule – to help the lucrative K&E company maintain a safe economical position from the encroaching computers, which were just then beginning to stir scientific imagination.

At MIT, after a year as instructor in the steam-engine laboratory, he rose to the level of associate professor of thermodynamics. Shortly thereafter he was appointed full professor.

His contacts with society were revived, briefly, when five colleagues in his department invited the lonely professor to be their dinner guest at the now-defunct, fancy, Smith House, a restaurant near MIT, to celebrate his new appointment. I worked there part time, earning a little extra money to help with tuition. As bus boy, I took care of the dirty dishes for that party. I can tell you what went on that night, so you'll know what kind of a guy the professor was.

Von Karlstein arrived at the restaurant on time, trusty slide-rule sticking out of the side pocket of his wrinkled blue serge suit. Sitting down, he took out the wooden instrument and laid it gently at his feet where it remained, like a panting Doberman pinscher fawning on its master. The professor recognized me in spite of my short-cut red jacket and red bow-tie.

"Fhat you doing here?" he asked.

"I work part time. Need the money," I answered.

"Goot boy," he said.

When the other guests had all arrived, von Karlstein, I remember, told everyone how he was brought up in the shadow of the Rhine Valley vineyards and how he considered himself the best expert on wines. Therefore, he said, the right to select the wine label belonged to him. Everyone agreed to letting the professor demonstrate his expertise in this matter. Me, I just watched and didn't say a word.

Once they were all seated around the table, the professor summoned Joe, Smith House's wine steward, with a snap of the fingers and a loud clucking of the tongue, explaining that this was the usual method on the continent for summoning wine stewards. It was the same way I used to call chickens at feeding time on my uncle's farm.

"Yah?" said the wine steward, a haughty creature who shuffles around in dark, dank cellars and comes up at dinnertime to make a sale.

"Fee," said the professor slowly, accenting each word. "Fee fill haff Sharpfein Weinfleglin, 1920 or 1921."

"D'only stuff we carry," said Joe icily. "Iz whad you seed in dis year book." He was angry. Wine stewards, who spend their lives in cellars, are not exposed to the sunshine of liberal education. Their minds are filled only with vintners, chateux-labels, dates, premier years, bouquets, colors and prices. Their life centers on their collection of bottles. Many a steward has been known to sit up through a long night, a sick bottle in lap, singing and rocking it back to recovery. And many a wine steward's career has been cut short from accidentally depositing on the dinner table a bottle of wine that had not completely recovered. Furthermore, wine stewards are temperamental and get angry very quickly. The best way to get them real mad is to ask for a bottle of wine they don't carry – which was what the professor had just done.

Joe opened a velvet-covered gold-engraved wine list, curtly dropped it on the professor's lap, and patiently looked up at the ceiling while twirling a big brass cellar key on chain. The professor studied the offerings for a good five minutes without saying anything.

"I recommend French Bordeaux Chateau Henri Louise, 1918," snapped the steward very impatiently. Nobody really reads a wine list for five minutes. Joe once told me that if somebody studies a wine list too long they either got bad eyes or they're trying to get him mad. "Christ's Sake," said Joe to me last year. "If you're looking up a phone number you don't spend five minutes studying a name."

"Please," said the professor calmly, shaking a finger at the wine steward. "Fee choose our own Cherman fine! Yes?"

He continued to explore the wine list a few more minutes

extensys corporation

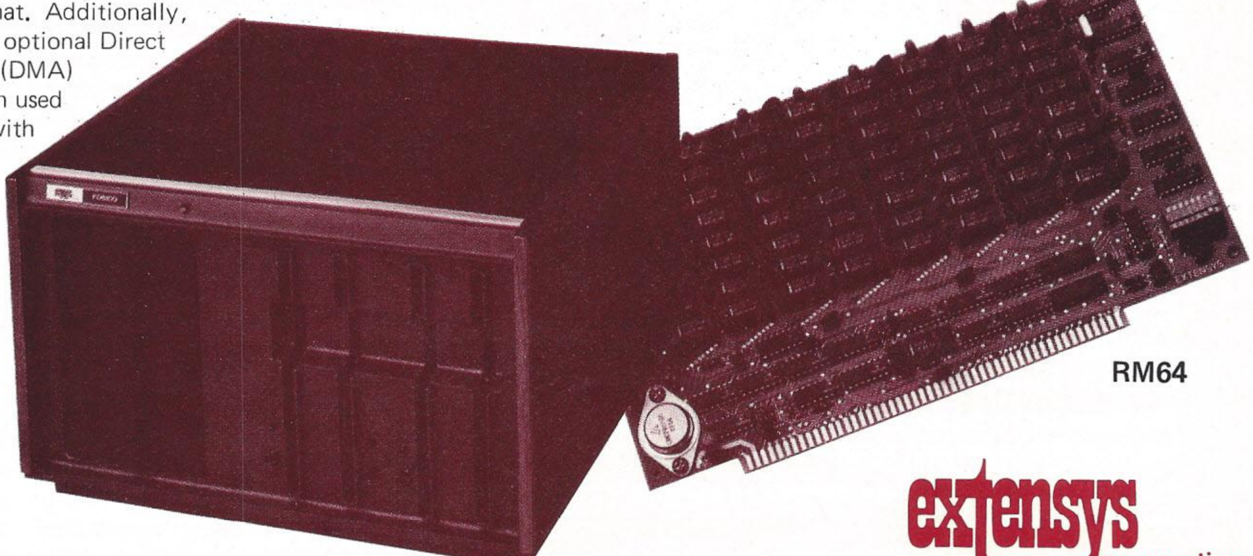
High performance, cost effective products for your microcomputer.

The Extensys FOS100 Floppy Disk System provides a high performance floppy disk system based upon multiprocessors in a multiuser environment. The FOS100 accomplishes this through a completely integrated hardware and software system. The hardware consists of a File I/O board, a PerSci controller board and one or two dual PerSci Model 277 floppy disk drives. The File I/O board has an 8080A microprocessor, 1K of PROM and 8K of RAM on-board to provide a separate I/O processor which will handle all disk I/O functions of the system. The PerSci controller board is encased in an attractive cabinet with either one or two dual PerSci drives.

The software, Extensys Multiprocessor Operating System, EMOS, provides many large computer capabilities, such as multiprocessor, multiuser operation, individual file security based upon user supplied passwords, system security and IBM compatible format. Additionally, the FOS100 has optional Direct Memory Access (DMA) capabilities when used in conjunction with the Extensys MM16 Memory Manager board.

The Extensys RM64 Memory Board provides up to 64K on one board. Designed for S-100 based computers, the RM64 occupies only one bus slot. Four versions of the same board are available: 16K, 32K, 48K and 64K bytes with the less-than-64K versions easily expandable by inserting memory chips into existing sockets. The Board uses dynamic RAMs with 300 ns access time and "invisible" on-board refresh. Other features include bank select logic for PROM/ROM coexistence plus expansion beyond 64K, write protect circuitry, and multilayer ground plane construction for noise immunity.

The RM64 comes completely assembled and tested with a one year warranty. Delivery is STOCK to 15 days upon receipt of order, with shipping and handling prepaid in the continental United States.



FOS100

RM64

Are you planning to produce a high performance, cost effective computer system or expanding the capabilities of your current line? If so contact US, the people who can make it happen for you NOW.

extensys corporation
Extensys Corporation
380 Bernardo Avenue
Mountain View, CA 94040
(415) 969-6100

while Joe kept muttering under his breath. Finally von Karlstein made his decision.

"Liebmilch Scheingebber, 1921, if you please. But von moment. I suckchest in future you haff Sharpfein Weinfleglin. Finest fine in der furld. Fee giff it allfays to aller Kleinkindern before schlafen gehen."

"Yah, sure," said Joe coldly to the professor. He went and fetched a bottle of Liebmilch, which happened to be the house wine. After passing around empty, sparkling tulip goblets, the steward put the bottle between his legs and in a *coup de grace* pulled the cork, showed it triumphantly to all the dining room, handed the cork to the professor, and poked the bottle into a bucket of ice. With proper motions he indicated that the professor was to fan the cork under the nose – not just stare at it.

"Fine, fet cork," said von Karlstein after a quick fanning ceremony, the tip of his nose twitching nervously now, like a bloodhound's just given a scent. Cork smelling is really a French trick. The Germans usually test their wine with less pomp – simply taking a swig right out of the bottle.

The steward wrapped a diaper around the wet bottle and poured a dash of wine into a clean, sparkling glass. With some fancy flourishes and a ceremonious rotation of the glass he handed it to the professor. Von Karlstein made several fancy flourishes of his own with the goblet, causing the wine to swish properly around in the glass and give up its bouquet. The professor performed this maneuver very well and his admiring colleagues gave him a polite round of applause, as though he had just served an ace on a tennis court.

Face flushed with pride, the professor acknowledged the plaudits with a gentle nod of the head. He also nudged the dozing slide rule with the toe of his boot as though to say to it: "You hear, mein friend? You hear, now, that they say about your Heinrich?"

He slowly raised the glass to his nostrils, fully distended by this time, and sniffed the bouquet. Instantly he let out a powerful sneeze. The blast blew the wine from the goblet and sprayed his hosts. Grabbing a napkin, he dabbed at his dripping face as did the others. He apologized to everyone, then blew his nose into the napkin, causing some diners in the expensive restaurant to drop their Strasburg pate. "Gut bouquet," muttered the professor, tears in his eyes.

Joe, who had remained tolerant throughout this exhibition, now produced a second goblet and with a disgusted look on his face he poured fresh wine into it. He handed it again to the professor. In an attempt to hurry up the slow-moving professor, the steward now brought an imaginary glass to his own lips, and smacked them loudly against each other in the manner of an impatient father urging a stubborn baby to eat cereal out of the spoon. The steward was becoming restless. Other diners were already snapping fingers at him and the professor had yet to take the first, customary, critical sip.

Von Karlstein, irritated by the hostility of the wine steward, abruptly turned his back to him, letting him know by this gesture that he didn't like his attitude at all. Then, with little finger stretched out at right angles to the goblet, the professor raised the delicate crystal to the lips and paused for a minute. "I fish to drink small toast," he said. "To the Keuffel-Esser Gleitenlineal. No computer fill efer take place from slide rules!" With a vengeance he poured the Liebmilch down the throat. He was at once seized by a fit of violent

coughing and wheezing. It became so bad that his head fell to the table and his chest heaved visibly as he gasped for air. The steward, who was familiar with such occasional paroxysms to his wine, gave the professor a real hefty whack on the back with more force than he usually applied to other diners. He was mad, I can tell you. When von Karlstein regained his breath a little, he looked around at his hosts. They were watching him apprehensively, waiting for his first words. In a barely audible voice, hoarse and strained, the professor groaned to the steward: "Ferry fine! Gut! Mein compliments!"

It was the last time his colleagues invited him out to dinner and it served further to embitter him toward life.

A hard taskmaster in class, von Karlstein began to drive us mercilessly. Occasionally, and probably because of a memory of his country's defeat in the war, he would spring surprise attacks on us. Unannounced quizzes. Most of us flunked.

The heated controversy of the times: clumsy, patched-up computers that were always breaking down versus those neat, lovable sliderules that could be monogrammed in gold.

One student, a sickly, scrawny, scared, near-sighted youth who spoke in a German accent like von Karlstein's, always got the highest grades. His name was Johann Glitzermann and it turned out that he was the professor's cousin. The professor was obviously proud of him and would hold him up as a model student to the rest of us.

On occasion, though, Johann, in his over-confidence, would screw up a simple demonstration problem at the blackboard and Professor von Karlstein, personally embarrassed, would scream, "You fill be der death of me yet, Johann! Fait!" And he gave his cousin a sharp clout on the head with the brass edge of the slide rule.

At semester's end, cousin Johann had a fat B-plus; the rest of us were lucky to escape with C-minuses. Von Karlstein, the way we figured it, wouldn't give an "A" to anybody, even if it was his own father and even if it was his father's dying wish. Von Karlstein's soul, someone said, had been twisted in a pretzel factory where he had worked before leaving Germany for America.

His work on slide rules brought him to the attention of the Keuffel and Esser company, which granted him money. He was to raise the slide rule's accuracy beyond three-place readings. His goal was to increase it to 6 places, a desirable precision when working in thermodynamics where small, fractional errors could lead to blow-ups in steam-engine design. This objective was the root of the heated controversy of the times; clumsy, patched-up computers that were always breaking down; versus those neat, lovable slide rules that could be personally monogrammed in gold.

Von Karlstein's first major contribution under the grant

was to add a three-power magnifying glass for better sighting of the fourth place. Later, he divulged that hairs from human chests were about to be placed between standard logarithmic graduations! Many cups of tea were dropped at MIT faculty meetings when the professor first revealed his novel idea.

He supported this theory from previous research. He had determined that finest lines for measuring instruments came from hairs on the human chest and armpits. Had something to do with filament cornification and density. The professor, who had a crop of hair on his chest that looked like a fur rug, willingly donated himself to the project. Every night after regular school hours, he could be seen bent over a work bench in the instrumentation laboratory. With sharp splinter forceps he would, from time to time, open his shirt and rip a single ripe hair from his chest. Each time, a muffled yelp escaped from his throat and tears spilled out of his eyes. The curious students peeking in had to close the door hurriedly, to hide the sound of their hysterical giggles. If he heard them, the dedicated professor gave no indication. His work was too important to be disrupted by a small gang of MIT hoodlums.

Using the forceps, he dipped the extracted hair into acetone to dissolve excess deodorants and body colognes, both of which he used freely, and then into a bath of gum arabic. Working under a large magnifying-glass stand, he then carefully placed the hair between the imprinted black lines, thus forming new sub-graduations. That done, he let out a pathetic sob and reached for another hair.

The project came to a halt when the professor ran out of hair and his chest was completely denuded except for a rash of small welts that took months to heal.

Keuffel and Esser, who were kept informed of his progress, liked the idea and told him to pursue it. Chest hairs on slide rules sounded like a good selling point to young, virile engineering students, they said.

Von Karlstein designed a gigantic slide rule, 300 feet long, to be built on the old MIT football field on Vassar Street.

Finally he produced a four-place dependable slide rule, but he wasn't satisfied. He still had not achieved a more accurate readout to six places.

At about this time, in 1935, scientific journals reported that Harvard was moving along well with its crude, high-speed calculator that threatened to replace slide rules.

Shaken, Keuffel and Esser anxiously contacted Professor von Karlstein, increased his monthly grant and urged him to intensify research. The company felt that if better calculating devices were possible, they would have to come from

the slide rule itself, the only calculating instrument to emerge since the bead-sliding abacus. It was the professor's obligation, said the company, to unlock any hidden potentials of the K&E instrument.

After lengthy meditations lasting far into the night, weeks at a time, von Karlstein finally had an idea. He designed a gigantic slide rule, 300 feet long, to be built on the old MIT football field on Vassar Street. The athletic society, which was using the field for underhand softball games with the girls from Simmons, opposed the spectacular idea, and they were supported by the administrative office. More lengthy self-deliberation followed. The professor finally unveiled a new "sectional-tongue slide rule." This proposal was the same 300-foot-long logarithmic scale he had first envisioned but now split into smaller sections, each 30 feet. They were arranged on a bed of travel so that activation of one would stimulate another. The giant slide rule was set up in the basement beneath the materials-testing laboratory.

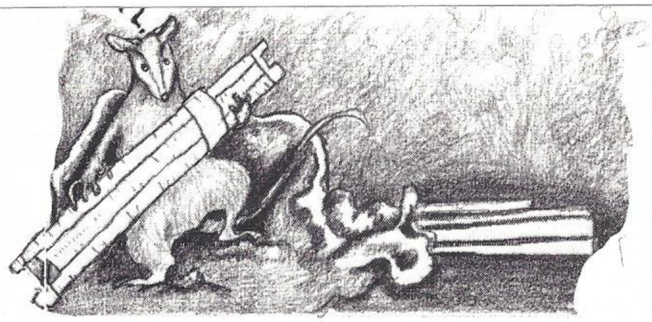
The colossus was to consist of 17 1/8-horsepower motors, 200 relay switches with three miles of connecting electrical wires, 300 feet of perfectly squared steel plates and 2,000 roller bearings. The peripheral slides were connected to the master board through paddle switches that permitted the slides to be moved quickly, gently and precisely to the proper logarithmic stop.

Von Karlstein received a new grant of $10,000 from IBM to complete the work. IBM, in return, was given the rights to construct and market the machine in the U.S. for Keuffel and Esser. The professor received consulting assistance from the Department of Mechanical Engineering, the Department of Electrical Engineering and the Department of Optics. The latter department designed the critical readout system: special micrometer mounted under microscope to give visual reading to five places with good approximation to the sixth.

One of the slide rule's peripheral plates fell on the professor's foot, broke his big toe and left him limp for life.

Several accidents occurred during construction. One of the peripheral plates fell on the professor's foot, broke his big toe and left him with a limp for the rest of his life. The 5,000 RPM motors overheated one day during a test run, burned out a paddle switch and went out of control. Two plates jammed, buckled in three places and required a month's delay before they could be replaced. Furthermore, the screeching of these noisy, high-speed motors damaged the professor's otic ganglion and produced 100% deafness in the left ear plus 60% loss in the right.

"*Why* do you do this?" asked Professor Douglas of the



Mathematics Department. "Why are you driving yourself?"

"Fee do not know fhy fee do fhat fee do," said von Karlstein cupping a hand to his ear in the direction of Professor Douglas. "Das leben ist Eine Grosse Sonderbar Fart!" – (life is one great, strange journey) – "und somethink there iss insite fitch drifes me. It fill be der death of me yet, fait! Fee seinen alle Ubermeistern von unserer eignen toten!" He then limped away leaving Professor Douglas to ponder on the philosophy he had pronounced.

The giant electrified slide rule, publicized as "biggest slide rule in the world" by K&E, required a full year to design and construct. When completed it filled the two small storage rooms allotted it and overlapped into an adjoining toilet through a 3' x 3' cut-out in the wall. This permitted peripheral #10 to be extended full length when operational.

The latter procedure, innocent but vital, finally brought the cumbersome slide-rule computer to an early end, although its doom was already forecast by eminent scientists who were watching developments at Harvard. Johann Glitzermann, whiz-kid cousin of the professor, was responsible for the sudden demise, though intrinsic safeguards could have prevented the accident which happened one night in the Fall of 1939.

Research engineers from all over the United States were coming to MIT to study the new device and employ it in their lengthy calculations. Time allotments had to be set up to cope with heavy traffic. Johann Glitzermann was assigned the 4-to-5 Wednesday night shift in the slide rule laboratory. On Wednesday night, November 5, 1939, while working on a lengthy thermodynamics Carnot-Cycle calculation, Johann made an error. He had been manipulating peripherals back and forth, at a rapid rate, along the steel tracks, when he accidently pulled the red "extend" switch and the adjoining green "retract" switch both at the same time. The machine, unable to travel in two directions simultaneously, stripped its gears. Peripheral #10, weighing 700 pounds, skidded free on its track, silently penetrated the opening in the toilet wall and pushed Professor von Karlstein into the men's urinal where he happened to be standing at that very moment. He was found with nose flattened against the wall, hand cupped to right ear, slide rule tucked under left armpit and torso squeezed permanently into the cavity of the urinal. As von Karlstein had predicted earlier, Johann Glitzermann had indeed been the death of him.

Professor Douglas of the Mathematical Department hurried down to look at his colleague when he heard the news. "We shall all be architects of our own destruction!" he intoned, repeating von Karlstein's earlier presentiment.

A blanket of passion flowers was laid on the bier, and the professor (still tucked into the urinal) was buried in a Lutheran cemetery near Radcliffe College, with the MIT chief plumber serving as assistant to the funeral director.

The giant electrical slide rule has long since been abandoned. But von Karlstein's vital research, which expanded limits of the slide rule and introduced electrical and optical assists, hastened the dawn of the computer age and made possible the conversion of mechanical computation to electrical. His small contribution to the development of the modern computer lives on in the quiet chronology of computer history.

Not long after the destruction of von Karlstein's slide rule, Harvard unveiled its Mark I Computer in time to be used in World War II for deciphering German codes. Mark I was a clumsy, over-heating hodgepodge of mechanical counters patched together from discarded business-data-processing machines. One thing Mark I proved, as Professor von Karlstein had predicted, was that the electric computer was indeed faster than a slide rule; but it would forever make mistakes and break down. It is a basic proposition in electricity that resistors, heart of the computer, are like candles that must eventually burn out. Computer repair-mechanics are in greater demand these days than are programmers, because of that resistor fault.

Although slide rules have been permanently displaced by the computer, it is safe to say that the computer would never have been developed without the slide rule. People like von Karlstein, real or unreal, have made the modern calculating-computer possible.

A small comet once burned briefly in the dark skies and left von Karlstein's trail of red ash among the stars, followed by his trusty slide rule, wooden tongue hanging out – both floating faithfully and forever along in the same orbit.

Videodisc Systems

BY LUDWIG BRAUN

What's your favorite movie? A year from now at your local hi-fi store, you might be able to buy a "record" that contains not only the soundtrack, but also the moving pictures. "Records" that hold pictures – not just sounds – are called "videodiscs." When you pop them into a drive attached to your T. V., your favorite movies appear on the screen.

What if you attach the drive to your computer instead? Prognosticators and pranksters have speculated; but here at last, in Professor Braun's report to the National Institute of Education, you get facts.

Videodisc recording goes back to 1927, when John L. Baird engraved video signals on a phonograph disk, in much the same way as audio signals. But since Baird was limited by the technology of his time to a bandwidth of 5,000 cycles per second, his pictures had only 30 lines and 15 black-and-white elements per line.

The next major step occurred in 1956, when Ampex announced recording video images onto magnetic tape. The major breakthrough here was the frequency modulation of the video signal prior to its recording. Because frequency modulation is independent of amplitude, the video signal essentially is recorded in two-level, or binary, form. Since then, several companies have worked intensively to develop videodisc systems. Among these companies are Telefunken-Decca, RCA, Philips, MCA, and Thomson-CSF. The developer of each system thinks its own is best. It probably will be several years before consumer interest in such systems is established, and before one system or another wins out over the others. (Unfortunately, all of these systems are incompatible with one another.)

Philips/MCA plans national distribution in 1978. Thomson-CSF plans to begin marketing in Europe in 1978.

There are essentially two kinds of videodisc systems: the electromechanical system in which the reading head touches the disc; and the optical one in which the only disc contact is optical (usually a laser beam). The RCA and Telefunken-Decca systems are electromechanical; the Philips/MCA and Thomson-CSF systems are optical. I'll focus specifically on the Philips/MCA system, because it will enter the American market first and because it appears to be superior.

In the Philips/MCA system, the disc rotates at 1,800 rpm. Each side has 54,000 tracks; each track contains a single frame of video information. Each disc side provides thirty minutes of video.

Because each frame occupies an entire track, you see one frame per revolution of the disc. So you can use the system as a slide projector in the stop-frame mode by jumping back one track at the end of each revolution; hence, the image stays still. You can also generate slow motion either forward or reverse by controlling the motion of the read head.

The system records the video signal in binary form by frequency modulation, as in the Ampex system. Here's what the information storage capacity looks like:

- You can store 54,000 individual slides on one disc side by using the stop-frame mode.
- If you compress the audio signal by a factor of 300 before recording and then "uncompress" by the same factor on playback, you can store 150 hours of music or other audio information on one disc side.
- George Kenney proposes a talking encyclopedia, containing 27,000 slides and 75 hours of commentary.
- Because the information is stored in binary, there is a total capacity of 185,625 bits per track, or 1.25 *billion* bytes per disc side! This enormous capacity is available as a read-only-memory for storing data and computer programs.

The *Encyclopedia Britannica* consists of 15 volumes, with about 1,000 500-word pages per volume, to give a total of 50 million characters of information. At one byte per character, the whole encyclopedia would consume only four percent of the capacity of a single disc. One disc probably could contain all the computer-based educational courseware which ever has been written.

In the Philips/MCA system, if the focusing arm is stationary, the tracking mirror can address any one of 100 tracks, and can move from any track to an adjoining one in 60 microseconds. The 100 tracks hold a total of 18,562,500 bits, or 2,320,312 bytes – 20 times the capacity of a Minifloppy disk. (See table below.)

The table reveals that the 100-track videodisc system is clearly superior to the Minifloppy disk system by at least an order of magnitude in capacity, transfer rate, and access time, with system cost, disc cost, and error rate being comparable in both systems. The only significant disadvantage of the videodisc is that it is read-only memory, whereas the floppy disk is read/write memory.

A potential drawback of the videodisc is production of copies. The repli-

Are videodiscs better?	PHILIPS/MCA videodisc, full capacity	PHILIPS/MCA videodisc, 100 tracks	NORTH STAR Minifloppy disk
Total capacity	1,250 megabytes	2.3 megabytes	.1 megabytes
Transfer rate	7 megabytes/second	7 megabytes/second	.016 megabytes/second
Cost per megabit	6/100 of a cent	40¢	$50
System cost	$500 + interface	$500 + interface	$700
Disk cost	$2 (in lots of 1,000)	$2 (in lots of 1,000)	$5
Average access time	5 seconds	.003 seconds	.463 seconds
Error rate	10^{-9}	10^{-9}	10^{-8} - 10^{-11}

cation equipment is sufficiently expensive that few schools will be able to justify the cost; however, both Philips and MCA are planning to establish replication centers at several locations. When the customer submits a videotape master encoded in NTSC format, the replication center will make a disc master for about $1,000 and make disc copies for $1 each; thus 200 copies will cost $6 each, and 1,000 copies will cost $2 each.

Educational applications

These educational applications are possible:

Linear video. Here, the videodisc replaces movie film. The cost of a 20-minute educational film is well over $200; even rental of the film for a single showing costs $15-25. The estimated purchase cost of a videodisc film is $10-15. So a school can build a "film" library for less than the present rental budget. In addition, the teacher can achieve fast forward and easy reversing, which are not available with movie film projectors.

Archival storage. Because videodiscs can store large capacities of information cheaply, they are an attractive medium for storing computer programs and data files, and for storing books, journals, and other printed materials. Because discs are compact and inexpensive, and because the information they contain is easily accessible, they are ideal for home and classroom use, as well as for community libraries.

Programmed instruction. Combining motion and still frames in full color with text and audio is more exciting than conventional programmed texts. The courseware developer has a much more flexible and powerful tool than he ever had with paper and printing press. Drs. Heustin and Bunderson and their WICAT group in Provo, Utah are developing an experimental videodisc programmed "text" in high-school biology for the McGraw-Hill book company, as well as conducting a study for the Mormon Church on the feasibility of using this medium for remote training of church officers stationed all over the world.

Interactive use with a computer. Dr. Alfred Bork of the University of California at Irvine suggests that a videodisc will contain a complete multimedia teaching package, which might include a color video sequence presenting historical background and other information to set the stage for a complex computer simulation of a phenomenon in biology, chemistry, or physics. After the video sequence, the disc would contain the simulation program, which would be loaded into the computer memory for subsequent execution. During interactive execution of the simulation program, the computer will call upon the videodisc to generate appropriate supporting graphics, an audio sequence, a background still frame, or even a new computer program. When the simulation is completed, the computer will call a testing program from the disc and check the depth of the student's understanding. Then the computer might call up another simulation (more or less complex, depending upon the student's performance) or a final video sequence.

One significant advantage to the teacher of this approach is the ease of use. Only a single disc need be loaded, rather than a computer program, a slide carousel, a film, and an audio cassette, each separately, and each into a different machine.

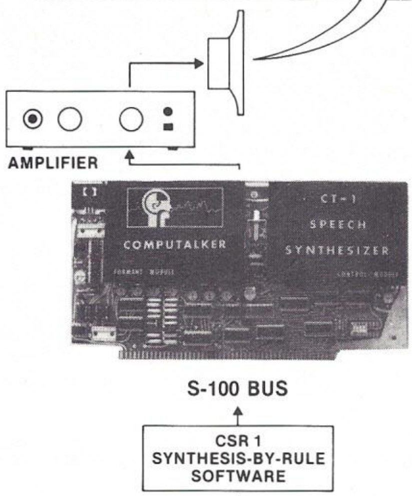
But the consumer videodisc player to be sold now will not be able to interact with a computer as indicated above. With that player, fast forward and reverse are executed under manual control. The track numbers are displayed on the screen during the search mode, but you must stop the player manually. A videodisc player must become available which can communicate directly with a computer – tell the computer which track it is reading currently and accept a computer command as to which track to read next. In this way, the player would be under direct and continuous control of the computer.

Philips and MCA are working independently on players for the educational and industrial users. These units are a year or two away, and are expected to cost in the order of $1,000-1,500. These so-called E/I players are expected to have a local microprocessor and local memory.

Because videodisc technology is so new (it still isn't quite here for most of us), little thought has been given to the inhibitions against using it. But we do have experience with the problems of introducing other technological devices, such as film projectors, slide projectors, audio cassette recorders, and computers. Based upon this experience, we may speculate that the main problems will be system reliability, teacher preparation, and available courseware.

This report was prepared at the request of the National Institute of Education, but any opinions are those of the author and don't necessarily reflect the policy or opinions of the National Institute of Education

COMPUTALKER



SPEAK "KAAMPYUTAOLKER"

MODEL CT-1 SYNTHESIZER	395.00
CSR1 SOFTWARE SYSTEM	35.00
DEMONSTRATION CASSETTE	2.95

CALIF. RESIDENTS ADD 6% SALES TAX

WRITE FOR INFORMATIVE LITERATURE

COMPUTALKER CONSULTANTS
P.O. BOX 1951, DEPT. D, SANTA MONICA, CA 90406

CIRCLE 49

We're sure you've heard how easy our **Music System** is to use and of its fantastic sound. At a suggested list price of $24.50 the **Music System** package for 8080 users includes: documentation, cassette tape (300 baud K.C. standard and 1200 baud CUTS) with the **Music System** in object form and six sample selections, and an S-100 bus circuit board kit with parts.

The **Music System** includes its own multifile editor, music generation routines and, most important, a high level music language compiler which support: key signatures, clef notations, whole notes thru sixty-fourth notes, triplets, staccato, articulation, accidentals, transposition and repeat capability.

Look for our **Music System**. Better yet, hear it at your local computer store.

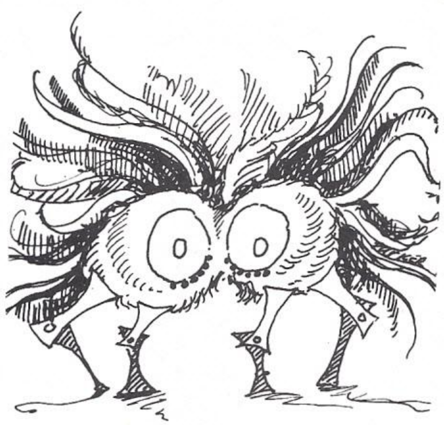
Software Technology Corporation

P.O. Box 5260 San Mateo, CA 94402

Prepaid mail orders only and we pay shipping within the U.S. CA residents add sales tax.

CIRCLE 50

Science Fiction



We're interested in publishing short science fiction articles with minicomputer central themes. Payment (consistent with length and quality) made upon acceptance. Must be original and not elsewhere published. Submit articles with self-addressed stamped envelope to **Personal Computing,** 1050 Commonwealth Ave., Boston, MA 02215.

CIRCLE 51

INQUIRE TODAY!!

We Ship From Stock

Business Systems Are Here Now!

COST EFFECTIVE HARDWARE • PERIPHERALS • SOFTWARE

Call our Mr. Amunson

Informative assistance in defining your objectives and choices.

For Engineers and Serious Hobbyists

BREADBOARDING • WIRE WRAP • DESIGN EQUIPMENT • TEST EQUIPMENT • COMPONENT PARTS • POWER SUPPLIES

• AP Products • OK Machine • Iso-Tip • B&K PRECISION • CONTINENTAL SPECIALTIES • NON-LINEAR SYSTEMS • PANVISE • VECTOR

Our 25th Year

ATLAS ELECTRONICS CORP

1570 3rd AVE. (88th St.) NEW YORK, N.Y. 10028
212-427-4040

Send $1 for complete set of catalogs (refunded on 1st purchase)

**We ship nationwide & overseas*

CIRCLE 52

The end of Kit-Kits.

The end of bad solder joints, heat damaged components and sick IC's. Introducing the Semikit. Item 1, a 16KRA Memory Board, $369.

Let's face it. Loading and soldering PC Boards is not much fun for the kit builder. Even more important, it's the place where most of the trouble gets introduced. The real fun and education comes in running and testing boards.

Now the Semikit with fully tested IC's.

At the price of a kit, Processor Technology Corporation introduces the Semikit. It's a fully stuffed, assembled and wave soldered PC Board loaded with IC's that have gone through Q.C. and final check-out (a first in the industry).

We leave you the fun of testing with our fully documented set of instructions. We do the production tasks of loading, wave soldering and inspecting the boards. You do the more interesting and time consuming chore of testing and burning-in the boards.

The result is one sweet deal for both of us. You get a board where the primary causes of damage (poor solder joints, excess solder and bad IC's) are virtually eliminated. You get a board of highest professional quality. And we get the business!

The 16KRA Memory Board's at your dealer now.

Your Processor Technology dealer has the first Semikit, a 16KRA Memory Board, in stock and ready to go right now. You can take it home tonight for $369 as a Semikit or for $399 fully assembled, tested and burned-in.

You'll have a 16,384 byte memory with a better price performance ratio than anything on the market today. Now you can afford to add quality, high density memory to your system for remarkably little. And you can add enough to solve complex computing problems right in the main frame.

The memory features invisible refresh. There's no waiting while the CPU is running. Worst case access time is 400 nsec. Each 4,096 word block is independently addressable for maximum system flexibility. Power is typically 5 watts, the same as most single 4K memory modules. Back-up power connection is built-in.

Other Semi's are coming your way.

The 16KRA Memory is Processor's first step in adding more fun, capability and reliability to your computer system at lower cost. Other modules are on the way to your dealer now. Come on down today.

Or you may contact us directly. Please address Processor Technology Corporation, Box L, 7100 Johnson Industrial Drive, Pleasanton, California 94566. Phone (415) 829-2600.

Processor Technology

CIRCLE 37

CLASSIFIEDS

Personal Computing Classified Rates

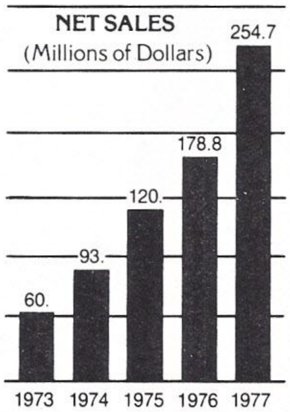
Individual: $1 per word; address set free; first two words set in all capital letters for free. If you want **bold** type, add $1 for each word set bold. Add $5 if you want replies sent to us and forwarded to you.

Company: 1X: $60/inch; 3X: $55/inch; 6X: $50/inch. Display ads accepted only if camera-ready materials submitted. Agency commission paid on camera-ready ads and after the first insertion of a publisher-set ad if subsequent insertions run without changes. Publisher sets ads in format and attempts to fit copy into space requested.

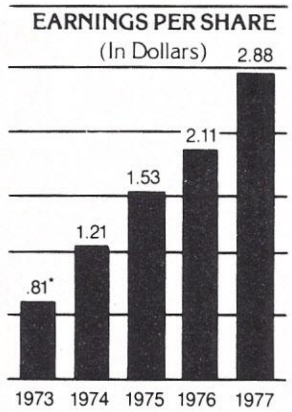
Advertisement accepted, however, with understanding that space may be larger than requested. Minimum billing size one inch; ads billed in half-inch increments thereafter. Frequency rate applies to publisher-set ads only if they run without changes in subsequent issues.

To place an ad, send copy to Classified Ad Dept., **Personal Computing,** *1050 Commonwealth Ave., Boston, MA 02215*

Is your career growing as fast as ours?



(1977 Fiscal Year)



*Adjusted to reflect a 3-for-1 stock split effected in the form of a 200% stock dividend declared in February 1973.

New Products Engineers (Analog and Digital)

Your responsibilities will include new product introduction of state-of-the-art processors, peripherals and controllers to our Systems Integration Division. A BSEE degree or the equivalent is required with a minimum of 3 or more years of related experience.

Diagnostic Programmers

You will design, code and debug assembler language programs for fault isolation, to the chip level, in digital systems. You will also write functional level system exercisers for stand-alone and disc-based real-time operating systems. A technical degree or the equivalent experience is needed, together with a thorough understanding of digital system hardware. Hardware troubleshooting experience and 3 or more years' programming experience, preferably in diagnostics is very desirable, especially with strong assembly language skills.

To apply for one of the above positions mail your resume to Tom Aldrich.

Test Engineers

You will be responsible for PCB test engineering support for digital products such as CPU's, memories and options. Experience is required in solving engineering problems as they relate to test equipment, diagnostics, test philosophy and component fault isolation. You will also initiate and implement improvements to the testing process and equipment, and provide coordination for the introduction of new products into production. A BSEE degree and two or more years' experience in PCB test or Test Equipment Design.

Manufacturing Engineers

You will be responsible for providing floor support in the PCB assembly area. A BSME, BSIE or the equivalent with 3 years' experience in Manufacturing Engineering is required. A knowledge of auto-insertion equipment assembly technique and soldering equipment is helpful.

To apply for one of the above positions mail your resume to Rene Santini.

Senior Systems Technicians

This position requires at least 3 years' experience, minimum, and an ASEE degree or its equivalent. Your background should include familiarity with CPU memory, moving head disks and related peripherals.

Special Systems Technicians

You will be working closely with the Special Systems Engineering Department in the design of non-standard computer products. A minimum of 3 years' experience and an ASEE degree or its equivalent is necessary.

To apply for one of the above positions mail your resume to John Prendergast.

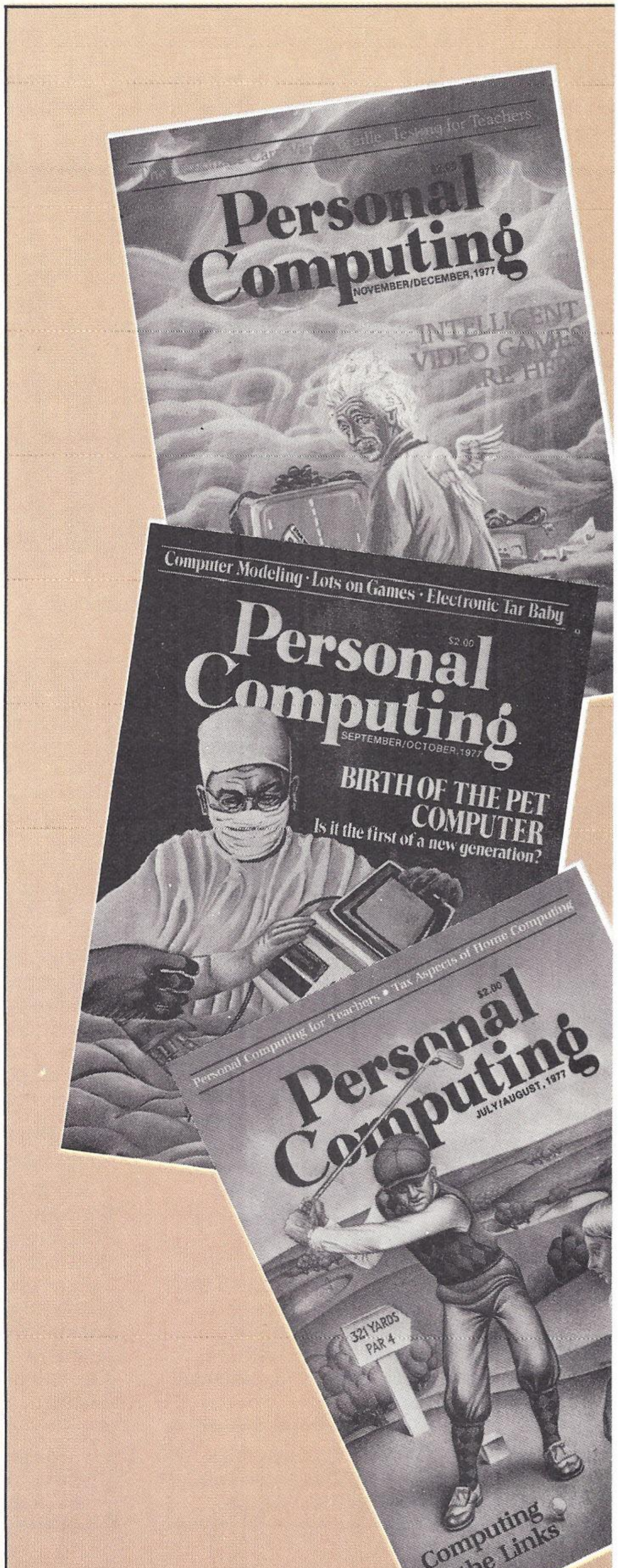
Data General Corporation, Route 9, Southboro, MA 01772. Data General is an equal opportunity employer, M/F.

Data General

ADVERTISERS' INDEX

SALES OFFICES

Northeast: Gary Dozier, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic**: Arthur Daks, 104 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Mid-West**: Hank Bean, 2633 Hillside Lane, Evanston, IL 60201, (312) 475-7161 • **San Francisco**: Ralph Peterson, 1380 Oak Creek Drive, #212, Palo Alto, CA 94304 • **West**: Carol Stagg, 823 Enchanted Way, Pacific Palisades, CA 90272; (213) 454-0624 • **Japan**: Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-Chome, Suginami-KU, Tokyo 166, Phone: (03) 311-1746.



How to Write for Personal Computing

Have you programmed your computer to converse in Gaelic? to do your home-ec homework? to read a bedtime story to the kids? Are you a frustrated fiction writer who's caught the computer bug? Or, have you found the ideal system or the absolutely worst combination of components?

Why not share your experiences with our readers? Yes, you too can write for *Personal Computing.* You choose the topic, *any* topic. If your topic relates to computers, great. If it relates to personal computers, even better. Computer hobbyists are looking for an excuse, any excuse, to buy a computer, and you might just offer the justification they're looking for.

We accept articles for all our sections – *Launching Pad* (our tutorial section for beginners), *On the Lighter Side* (where we print humorous applications), *In the Money* (how to use your computer to benefit financially), *Digging In* (for our more "advanced" topics), and *Once Upon a Time* (where we let your imagination run wild). We'd love to see some comparisons of computers or computer products. Tell us the good *and* bad of your system.

Keep your writing simple. No, our readers are not simpletons or beginners, but if you can explain something in simple words, do so. Don't clutter your piece with unnecessary jargon. If you're already into computers, give the newcomers a hand and let them in on some of the tricks of the trade – in simple terms. Examples, analogies, and charts and diagrams help both the beginner and the more advanced user appreciate what you're saying. Feel free to use "I" and "you" to make your article more personal and meaningful to the reader. Put the reader in the position of programmer ("you"). Also, please, please do not write your entire article in caps. And please indent for each paragraph.

Some things to note. Make sure your details are accurate – especially prices, other numerical information, and company names. Don't rely on hearsay or memory.

If you write about a program you've invented, try this order (to make sure you cover all angles): state the program's purpose; show a sample run; explain what the input options are, and what the output means; show another sample run; explain the underlying theory (if any); state the language, version, and computer you used and their peculiarities; show the listing; explain the program's over-all structure; analyze the program's details line by line; and suggest how the reader might improve or change the program.

Whatever your area of interest, you can turn it into an article. For example, if you're interested in watching birds then why not try an article on how to use a computer to track bird migrations? Or if your business is ________________, why not try a piece on computers and how they can be used to ______ ________________. We're open to ideas . . . .

If you've never written for a publication before and you'd like to discuss your piece with us before beginning it, give us a call. (Please do *not* mail us vague story proposals or outlines. We'd rather see the first few paragraphs of your article.) We'd be glad to discuss what you have in mind, and offer a few ideas of our own.

As a matter of form, we prefer (and are more likely to accept) articles that have been typed. Most of our articles run around 2-4 magazine pages. (There's about 3-3/4 typewritten pages to a magazine page.) But if what you have is good, we'll compromise and print anything from a sentence to your version of the encyclopedia.

Now here's the good part: we pay for any original material we print, although the price varies depending on the *quality* of the article. (So make it good!)

Why not give it a whirl? There may be a latent Hemingway, Fitzgerald or Asimov beneath that Einsteinian exterior.

FURBUSH MANOR

BY LAURANCE F. WYGANT

Last Wednesday I accepted a dinner invitation from my old friend Thomas A. Furbush – "Furry", we used to call him (for his face, not his name, of course). The only exception was our Logic 101 instructor, who always called him "Little Fuzzy". I thought it touching that old Dorchester was such a fan of H. Beam Piper, until I got to know Furbush better. Dorchester took early retirement the year after Little Fuzzy graced his classes.

Anyway, I went to Furry's. I wanted to see his highly touted micro system as much as his wife's lasagna. Furry had his whole house wired. He spent months on it, he said, but it was worth it. He grinned when I asked to see it, and whispered, "He's everywhere!"

Furry took the Chianti and put it in a rather lumpy-looking empty ice bucket, explaining about the precisely-controlled refrigeration coils. He dialed 50 degrees, earning a black look from me. Apologetically, he promised an exact 68 temperature. His machine was aware that people often tried to drink their wine too chilled, and compensated.

We got cocktails – uncomputerized – and wandered over to Furry's keyboard and video display. Furry explained that the reason his TV set was upside down was that it was easier than rewiring the video board, and anyway it was a portable set, so it was simple enough for the kids to turn it over to see real TV. I said I thought the right-to-left display looked a bit odd, but Furry claimed it was just a matter of getting used to it – and besides, it broke the monotony, rather like a game really. I allowed it would certainly lend interest to interaction.

During dinner he told me about his personal finance package and his menu application. He claimed to be moved strongly toward marketing both – since the bankruptcy proceedings had been completed, the finance package had run fine, and the menu package worked very well, although the sardine croquettes had gotten tiresome during debugging. I admired the menu program – Hollandaise sauce on the lasagna was novel, and I would have never thought to try meatballs in the asparagus myself – a very creative piece of software.

The only discordant note was when Sam, the machine, announced, "Coffee's ready" in a stereo soprano. Ellen positively cowered! Furry looked unhappy. He mentioned briefly how he had at first had a slight wiring accident and connected the coffeemaker to the automatic fire-alarm call-in unit, which got crossed with the burglar alarm call-in. That night the whole police and fire departments came over for coffee. Fortunately no one got shot, the foam didn't reach Sam-the-computer, and they let Furbush out of jail in the morning.

Furry was quite proud of his auto-call units. He said he'd had no end of grief debugging them, but fortunately nobody ever managed to trace the rash of obscene midnight phone calls in North Carolina to Furbush in Chicago. The calls to Tokyo were paid off finally, and things seemed to have settled.

As I got up to leave, I said good-bye to Sam. Sam said, "Fire phasers! Warp 9!" I strolled down the walk, stepping aside to let the firemen past and waving to the police sharpshooters in the hedge.

We Have GOOD NEWS and BAD NEWS

The good news is that you can now add any S-100 bus compatible component to The VERSATILE CRT.

The bad news is you'll have to decide for yourself what components will combine to make the best system for you.

Just by looking you can see it's a rugged, professional unit with a 9'' video monitor covered with smoked plexiglass, and a 53-key ASCII keyboard.

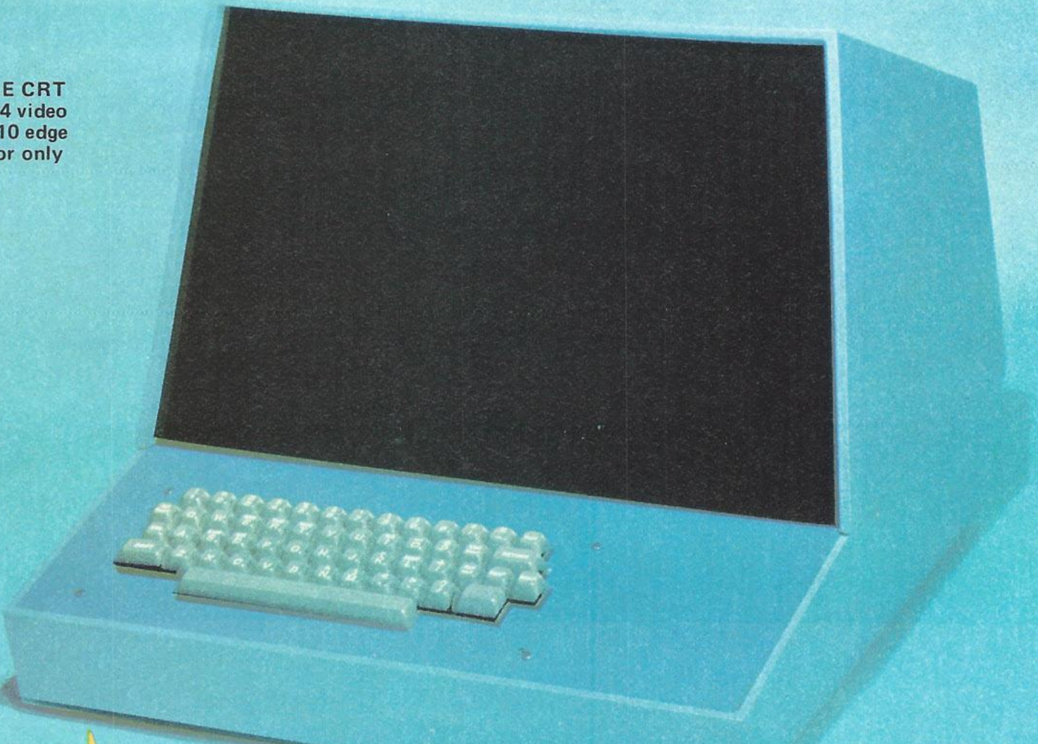
But there's more than meets the eye. What you can't see is the mainframe which is a 10-slot plexiglass card rack. Or its fully shielded motherboard with 10 spaces provided for sockets (2 sockets are already included for you). Or the heavy duty filtered power supply and 75 CFM cooling fan. Or the power switch and cord on the rear panel. Or the space available for a floppy disk system. Or the expansion capabilities on the back panel for the addition of sockets.

You'll receive The VERSATILE CRT fully assembled with a 90-day warranty, about 20 days after we get your order. Or you can buy one at your nearest dealer. Dealers can have super fast delivery from stock to 20 days. You'll also receive complete documentation and operating manual. ~~Total price for The VERSATILE CRT is $699.95~~.

Dealer inquiries are welcomed.

Compact and Expandable, The VERSATILE CRT lets you do it your own way.

NOW!
The VERSATILE CRT
includes a 16x64 video
display card & 10 edge
connector all for only
$999.95



COMPUTER DATA SYSTEMS

5460 Fairmont Drive • Wilmington DE 19808
(302) 738 - 0933

CIRCLE 2


System 8813